



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**KONVOLUČNÍ NEURONOVÉ SÍTĚ NA PLATFORMĚ
WINDOWS**

CONVOLUTION NEURAL NETWORKS ON THE WINDOWS PLATFORM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Martin Kapusta

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Přinosil, Ph.D.

BRNO 2017

Bakalářská práce

bakalářský studijní obor **Teleinformatika**

Ústav telekomunikací

Student: Martin Kapusta

ID: 170606

Ročník: 3

Akademický rok: 2016/17

NÁZEV TÉMATU:

Konvoluční neuronové sítě na platformě Windows

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte teoretické možnosti nasazení konvolučních neuronových sítí na platformě Windows. V rámci práce provedte instalaci vybraného frameworku pro tvorbu a trénování konvolučních neuronových sítí na operačním systému Windows. Dále navrhnete a realizujete nástroj pro automatickou instalaci daného frameworku včetně všech potřebných knihoven na osobní počítač s operačním systémem Windows 7 nebo 10. Ověřte správnou funkčnost vytvořeného nástroje na několika různých počítačích.

DOPORUČENÁ LITERATURA:

[1] HÁLA, Pavel. Klasifikace spekter pomocí konvolučních neuronových sítí. 2014. PhD Thesis. Masarykova univerzita, Přírodovědecká fakulta.

[2] JIA, Yangqing, et al. Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014. p. 675-678.

Termín zadání: 1.2.2017

Termín odevzdání: 8.6.2017

Vedoucí práce: Ing. Jiří Přinosil, Ph.D.

Konzultant:

doc. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Cieľom bakalárskej práce bolo naštudovanie najnovších poznatkov z oblasti konvolučných neurónových sietí a ich použitia. Práca popisuje históriu ich vzniku, biologický neurón a analogický matematický model neurónu. Tiež oblasti, kde sa neurónové siete používajú ako aj oblasti do ktorých sa postupne rozširujú, spôsoby učenia, rozdiely medzi konvulčnými neurónovými sieťami a klasickými neurónovými sieťami a ich architektúrou. Práca sa skladá s dvoch častí. V prvej je popisovaný výber frameworku pre prácu s konvulčnými neurónovými sieťami, ktorý je vhodný pre implementáciu do operačného systému Windows, inštaláciu zvoleného frameworku a riešenie problémov s danou inštaláciou. Druhá časť je zameraná na vytvorenie nástroja pre automatizovanú inštaláciu Caffé frameworku pre operačný systém Windows 7 a Windows 10, vytvoreného v jazyku JavaFX.

KLÚČOVÉ SLOVÁ

Neurón, konvulčná neurónová sieť, Caffé, inštalátor, Windows, platforma CUDA

ABSTRACT

The aim of the bachelor thesis is the latest knowledge of convolution neural networks and their application. The thesis describes the history, biological neuron and analogous mathematical model of a neuron. It also deals with the areas where neural networks are used, as well as the areas in which they expand gradually, the ways of learning and training, the differences between convolution neural networks and classical neural networks and their architecture. The thesis consists of two parts. The first part is the selection of the framework for working with convolution neural networks, which is suitable for implementation in the Windows operating system, the installation of the framework and its troubleshooting. The second part is aimed at creating an automated installation tool for the Windows 7 and Windows 10 operating system, created in JavaFX.

KEYWORDS

Neuron, convolutional neural network, Caffé, installer, Windows, platform CUDA

BIBLIOGRAFICKÁ CITÁCIA:

KAPUSTA, M. *Konvoluční neuronové sítě na platformě Windows*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2017. 59 s. Vedoucí bakalářské práce Ing. Jiří Přinosil, Ph.D..

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Konvoluční neuronové sítě na platformě Windows“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

podpis autora

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi Ing. Jiřímu Přinosilovi, Ph.D, za odborné vedenie, trpezlivosť, konzultácie a cenné rady pri spracovaní bakalárskej práce.

Brno

.....

(podpis autora)

OBSAH

1	HISTÓRIA VZNIKU NEURÓNOVÝCH SIETÍ	12
2	NEURÓN	13
2.1	Biologický neurón.....	13
2.2	Umelý neurón – matematický model	14
2.3	Neurónová sieť	17
2.4	Učenie neurónovej siete.....	19
3	KONVOLUČNÉ NEURÓNOVÉ SIETE	21
3.1	Základné informácie	21
3.2	Architektúra konvolučnej neurónovej siete.....	22
3.3	Učenie konvolučnej siete	24
3.4	Oblasti používania konvolučných neurónových sietí.....	24
4	INŠTALÁCIA PRE PLATFORMU WINDOWS	27
4.1	Dostupné frameworky.....	27
4.2	Výber Frameworku	28
4.3	Úvod k inštalácii	29
4.3.1	Open Source licencia.....	30
4.4	Inštalácia po jednotlivých krokoch.....	31
4.4.1	Potrebný softvér	31
4.4.2	Postup inštalácie s použitím platformy CUDA.....	32
4.4.3	Postup inštalácie bez použitia platformy CUDA.....	40
4.5	Riešenie problémov pri inštalácii.....	42
5	NÁVRH NÁSTROJA PRE AUTOMATIZOVANÚ INŠTALÁCIU	43
5.1	Úvod k tvorbe inštalátora.....	43
5.2	Realizácia inštalátora	44
5.2.1	Príprava súčastí programu.....	44
5.2.2	Programovací jazyk JavaFX.....	44
5.3	Popis jednotlivých komponentov.....	45
5.3.1	Inštalátor Caffé s použitím platformy CUDA	45
5.3.2	Inštalátor Caffé bez použitia platformy CUDA.....	48
5.4	Popis inštalácie a testovanie	50
6	ZÁVER	52

ZDROJE	54
ZOZNAM SKRATIEK	57
ZOZNAM PRÍLOH.....	58
A OBSAH PRILOŽENÉHO DVD	59

ZOZNAM OBRÁZKOV

Obr. 2.1 Biologický model neurónu.....	13
Obr. 2.2 Model nervovej sústavy v mozgu	14
Obr. 2.3 Ukážka spojení neurónov.....	14
Obr. 2.4 Lineárna prenosová funkcia	15
Obr. 2.5 Skoková prenosová funkcia	16
Obr. 2.6 Sigmoidná prenosová funkcia	16
Obr. 2.7 Matematický model biologického neurónu	17
Obr. 2.8 Sieť s dopredným šírením signálu.....	19
Obr. 2.9 Sieť so spätnoväzobným šírením signálu	19
Obr. 3.1 Grafické znázornenie max-pooling algoritmu	23
Obr. 3.2 Príklad konvolučnej neurónovej siete s jednotlivými vrstvami	23
Obr. 4.1 Stiahnutie Caffé frameworku.....	32
Obr. 4.2 Ukážka stránky s rôznymi verziami CUDY	33
Obr. 4.3 Ukážka upraveného súboru CommonSettings.props.....	35
Obr. 4.4 Otvorenie „Manage NuGet Packages for solution...”	36
Obr. 4.5 Tlačidlo „Restore”	36
Obr. 4.6 Odištalovanie pôvodného OpenCV	37
Obr. 4.7 Ukážka odznačenia projektov	37
Obr. 4.8 Inštalácia OpenCV Default Build.....	38
Obr. 4.9 Výsledok inštalácie	38
Obr. 4.10 Ukážka nastavenia režimu	39
Obr. 4.11 Kontrola konfigurácie	39
Obr. 4.12 Výsledok kompilácie.....	40
Obr. 4.13 Finálne upravený súbor CommonSettings.props.....	41
Obr. 5.1 Inštalátor Caffé s použitím platformy CUDA	46
Obr. 5.2 Inštalátor Caffé bez použitia platformy CUDA	48

ÚVOD

V dnešnej dobe sa informatika čoraz viac zaoberá možnosťami vytvorenia umelej inteligencie, ktorá by našla využitie v mnohých súčasných oboroch. Cieľom je vytvorenie programu, ktorý by bol schopný samostatného učenia sa a konštruktívneho reagovania na rôzne situácie a podnety v komplexnom prostredí, teda aby pracoval bez presne určených krokov z externého zdroja. Bežná umelá inteligencia sa snaží dosiahnuť schopnosti vyššej nervovej sústavy pomocou deduktívnych algoritmov. Základom a princípom dnešných počítačov je presná lokalizácia informácie v pamäti a jej sekvenčné zapracovanie. Tento princíp však nefunguje pri všetkých úlohách umelej inteligencie. Mnohé úlohy sa nedajú algoritmicky riešiť alebo majú riešenie s exponenciálnou zložitosťou, ktorá rýchlo prerastá súčasné technické možnosti. Človek tieto úlohy úspešne rieši, preto sa obor umelej inteligencie prepája s biológiou a svoju pozornosť obracia na ľudské schopnosti. Zameriava sa hlavne na nervovú sústavu a jej hardvérovú a softvérovú realizáciu v podobe vytvorenia umelej neurónovej sústavy. [1]

Neurónové siete sa zaradujú do oboru umelej inteligencie presnejšie do časti zaoberajúcej sa hĺbkovým učením. Hlavným cieľom prečo sú neurónové siete vyvíjané a v posledných rokoch sa im dostáva stále väčšia pozornosť je, že dokážu vyextrahovať hlbší význam zo zadaných dát. Na základe získaných informácií sú schopné porozumieť abstraktným pojmom a tým reagovať na rôzne situácie podobne ako človek. [2]

Ak sa vložia do systému fotky rôznych zvierat, systém si určí základné tvary a špecifické tvarové vlastnosti všetkých zvierat a priradí si k nim názov, toto nazývame učenie neurónovej siete. Na základe naučených „schopností“ sa následne snaží rozpoznať špecifické prvky objektov na fotke, ktorú chceme analyzovať bez nutnosti naprogramovania konkrétnej podoby objektu. [3] Tiež prebieha výskum o možnostiach použitia neurónových sietí v oblastiach porozumenia významu reči, textov, pochopenia kontextu obrázkov.

V bakalárskej práci je popisovaná história vzniku neurónových sietí. Dôležité pre pochopenie ich princípov je potrebné objasniť si pojmy ako biologický neurón a

umelý neurón. Ďalej sú popisované najbežnejšie typy prenosových funkcií matematického modelu neurónu, štruktúra umelej neurónovej siete analogicky prirovnaná k nervovej sústave človeka, rozdiely medzi sieťami s dopredným a spätnoväzobným šírením signálu.

Nasledujúca kapitola popisuje konvolučné neurónové siete a ich jednotlivé vrstvy - konvulučnú, podvzorkovaciu a plne prepojenú. Podrobnejšie sú popisované rozdiely v učení oproti klasickým neurónovým sieťam a ich použitie v súčasnosti.

Ďalšia časť je zameraná na výber frameworku vhodného na inštaláciu pre operačný systém Windows 7 a Windows 10. Popisovaný je základný softvér potrebný pre inštaláciu, licencie frameworkov a stránky, z ktorých sa jednotlivé programy potrebné pre inštaláciu na operačný systém Windows, dajú získať. Nakoniec je vytvorený podrobný návod inštalácie Caffé frameworku pre operačný systém Windows s použitím platformy CUDA a aj bez použitia platformy CUDA, implementovaný do Microsoft Visual Studio 2013, po jednotlivých krokoch dopĺňaných obrázkami pre jednoduchšiu orientáciu pri inštalácii. Univerzálny návod bolo nutné vytvoriť kvôli tomu, že frameworky pre prácu s konvulučnými neurónovými sieťami sú určené pre operačný systém Linux a ich implementácia pre operačný systém Windows je náročná. Autori neposkytujú oficiálnu podporu pre operačné systémy Windows. Užívateľ je schopný podľa návodu nainštalovať Caffé framework pre operačný systém Windows bez problémov, nemusí inštalovať operačný systém Linux, riešiť problémy s inštaláciou pre Windows a môže sa sústrediť na tvorbu modelov.

Posledná časť sa zaoberá tvorbou automatizovaného nástroja pre inštaláciu Caffé frameworku pre operačný systém Windows 7 a Windows 10. Aplikácia je napísaná v programovacom jazyku JavaFX. Pomocou vytvoreného nástroja je užívateľ schopný implementovať Caffé framework do vývojového prostredia Microsoft Visual Studio 2013 bez nutnosti dlhých diskusií na fórach, hľadania informácií a riešenia početných problémov. Inštalátor pozostáva z krokov, ktoré som uviedol v univerzálnom návode pre inštaláciu, pretransformovaných do dvoch verzií inštalátora s použitím platformy CUDA a aj bez použitia platformy CUDA.

1 HISTÓRIA VZNIKU NEURÓNOVÝCH SIETÍ

História neurónových sietí sa datuje až k roku 1943. V tomto roku vytvorili Warren McCulloch s Walrterom Pittseom prvý umelý model neurónu. Neskôr v roku 1949 bola Donaldom Hebbom vydaná kniha s názvom „Organization of Behavior“ v ktorej boli navrhnuté pravidlá na učenie sa neurónov známe do dnešnej doby ako Hebbovské učenie. Martin Minský, skonštruoval prvý neuropočítač, viedol proti neurónovým sieťam diskreditačnú kampaň. Na jej základe vznikol článok „Perceptron“ v roku 1969 ktorý zdôrazňoval neschopnosť neurónov naučiť sa funkciu XOR. V roku 1982 John Hopfield – fyzik vydal dva články o algoritme Back-Propagation – spätné šírenie určené pre učenie viacvrstvových neurónových sietí. Prelomovým systémom v tejto oblasti bol systém NETtalk, ktorý umožňoval prevod anglického písaného textu do hovoreného slova. Dosahoval kvalitnejších výsledkov ako systém DECtalk, ktorý plnil rovnakú funkciu používaním zložitých algoritmov. [4] V roku 1982 sa tiež uskutočnila konferencia, na ktorej Japonsko uviedlo novú, piatu, generáciu snahy o vytvorenie umelých neurónových sietí. USA nechcelo byť pozadu vo výskume tak začali dotovať vo veľkej miere výskum v tomto obore.

Prvá generácia obsahovala spojenia pomocou vypínačov a vodičov. Druhá generácia pozostávala z tranzistorov. Tretia generácia bola tvorená integrovanými obvodmi a vyššími programovacími jazykmi. Štvrtá generácia je už primárne zameraná na programovanie vo vyšších programovacích jazykoch. V roku 1986 boli novinkou viacvrstvové siete. Vtedy vznikol problém s tým, ako rozšíriť Widrow-Hoffové pravidlo na viacvrstvé siete. S myšlienkou prišiel tím Davida Rumelharta. Táto myšlienka je podobná dnešným neurónovým sieťam so spätným šírením signálu – chyba sa distribuuje a koriguje prostredníctvom celej siete s mnohými vrstvami. Na rozdiel od hybridných, ktoré majú vrstvy len dve. Výsledkom bolo zhodnotenie, že neurónové siete so spätným šírením sú pomalé na učenie, pretože potrebujú tisíce operácií na učenie. [5]

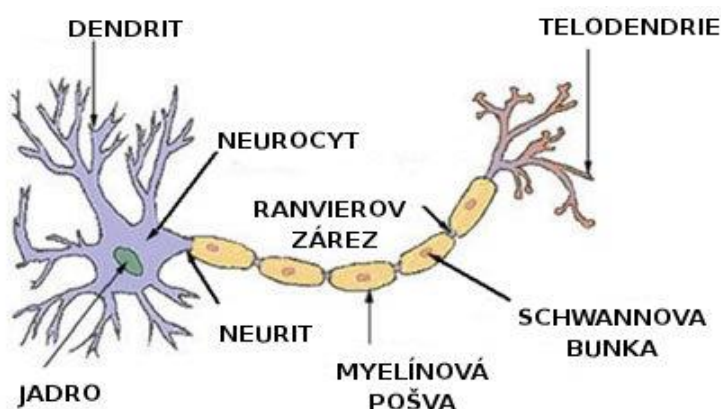
Budúcnosť neurónových sietí by mala byť vo vývoji hardvéru, ktorý by urýchl'oval niektoré operácie. Niektoré firmy sa snažia vytvoriť integrované obvody, ktoré by boli na účel používania neurónových sietí prispôsobené.

2 NEURÓN

Keďže sa daná bakalárska práca zaoberá neurónovými sieťami, tak je potrebné priblížiť si neuróny biologické, na základe ktorých funguje nervová sústava tela človeka a následne modely neurónov, ktoré sa používajú pre potreby umelej inteligencie.

2.1 Biologický neurón

Neurón alebo neurónová bunka je základná stavebná jednotka nervovej sústavy. Neuróny sú vysoko špecializované bunky schopné prijímať, prenášať a spracovávať signály z vonkajšieho aj vnútorného prostredia a podmieňovať organizmus reagovať na jeho podnety. Štruktúra neurónu je zobrazená na obrázku [Obr. 2.1] Prevzatý z [24].



Obr. 2.1 Biologický model neurónu

Skladá sa z jadra, v ktorom býva väčšinou jadierko. V cytoplazme tela sa nachádzajú organely. V tele neurónu a aj v jeho výbežkoch sa nachádzajú vo veľkom množstve mitochondrie a lyzozómy. Z tela (somy) vybieha niekoľko desiatok až stoviek krátkych výbežkov. Nazývame ich dendridy a jeden dlhý výbežok, ktorý nazývame neurit alebo axon. Dendridy sú dostredivé a neurity odstredivé. [6]

Telo neurónu má dĺžku niekoľko mikrometrov, dendridy majú veľkosť rádovo v milimetroch a axony môžu v niektorých prípadoch dosahovať dĺžku až vyššie jedného metra. Axony sú zakončené synapsami, ktoré sa spájajú s inými

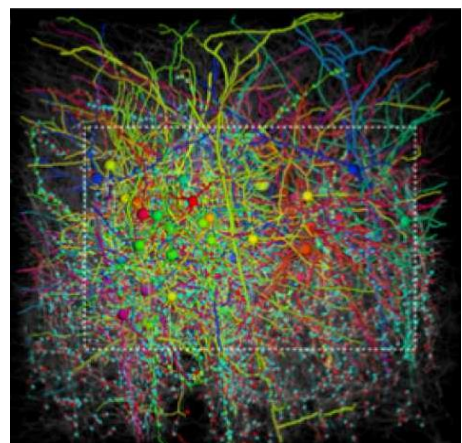
neurónmi, aby mohli medzi sebou viesť vzruchy a prenášať ich do mozgu. Každý neurón môže byť spojený až s 5000 ďalšími neurónmi. Na začiatku axónu prebieha sčítanie informácií. Ak sčítanie prekročí prah významnosti, axón začne šíriť akčný potenciál, ktorý funguje na elektrickom princípe. Akčný potenciál doputuje až k zakončeniu neurónu a vyvolá synapsiu - chemický dej, ktorým sa vzruch preniesie na ďalší neurón. [6]

Na prvom obrázku, [Obr. 2.2] [7], je model nervovej sústavy v mozgu človeka. Na svete neexistujú dvaja ľudia s rovnako poprepájanými neurónmi tieto spojenia sú jedinečné ako otlaky prstov.

Na druhom obrázku, [Obr. 2.3] [8], je najpodrobnejšia mapa spojení v mozgu, ktorá zobrazuje aktivitu mozgových buniek pri reakciách na rôzne stimuly a dekodovaní prijatých informácií. Daná mapa bola vytvorená skúmaním myší. U človeka by prepojenia medzi nervovými bunkami boli omnoho početnejšie.



Obr. 2.2 Model nervovej sústavy v mozgu



Obr. 2.3 Ukážka spojení neurónov

2.2 Umelý neurón – matematický model

Matematický model neurónu vykonáva približne podobnú funkciu ako biologický neurón kvôli tomu, že v súčasnosti nie sme schopní vytvoriť analogicky presný matematický model neurónu k biologickému, pretože neuróny sú veľmi zložité objekty a presný spôsob spracovania a prenosu informácie sa doposiaľ vedci v oblasti biológie snažia objasniť. Jeho skladbu však poznáme. Biologický neurón má dendridy, matematický má vektorové vstupy $\vec{x}=(x_1, \dots, x_n)$, ktorými získava dáta.

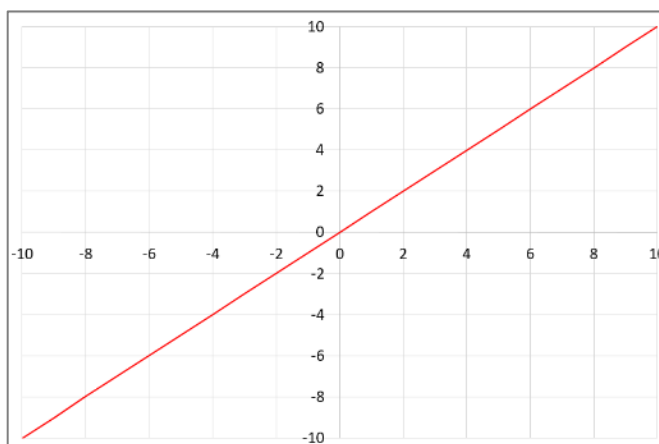
Tieto vstupy majú pridelené určité vektorové váhy $\vec{w}=(w_1, \dots, w_n)$. Výsledná suma zozbieraných informácií zo všetkých váhových vstupov vyvoláva aktivačnú funkciu $f(u)$, ktorá je zväčša, ale nie je to pravidlom, nelineárna. Obsahuje taktiež prah p . Tento matematický model zosilňuje alebo potláča vstupy a tým znižuje riziko vzniku chyby pri prenose informácie. Vstupy upravené váhami sa sčítajú spolu s prahom. Tento výsledok nazývame vnútorným potenciálom neurónu. [9] [10]

Základnú funkciu popisuje rovnica (1. 1).

$$ik = f(u) = f\left(\sum_{i=1}^k x_i w_i + p\right) \quad (1.1)$$

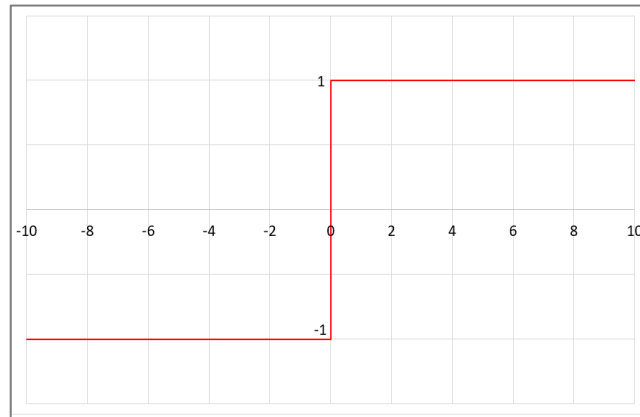
- k je počet vstupov
- x_i je i ty vstupný signál neurónu
- w_i je i ta vstupná váha neurónu
- ik je vstupný signál neurónu
- f je aktivačná funkcia neurónu
- u je potenciál neurónu
- p je prah

Výstup neurónu závisí od prenosovej funkcie. Ako príklad uvádzam tri základné funkcie. Najjednoduchšia prenosová funkcia je lineárna [Obr. 2.4], všetko čo príde na vstup sa bez zmeny prenesie na výstup. [9] $f(x) = x$



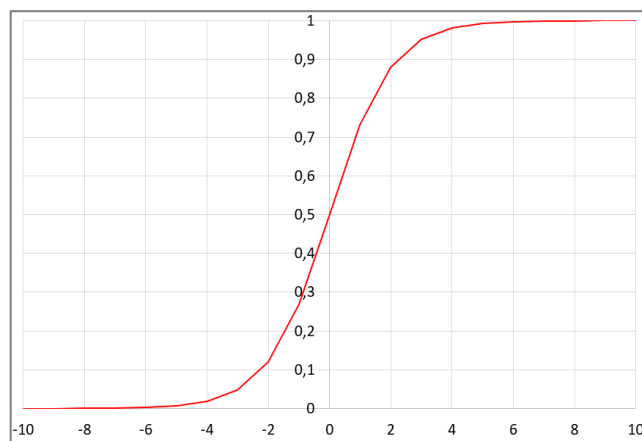
Obr. 2.4 Lineárna prenosová funkcia

Druhým typom je skoková funkcia [Obr. 2.5], ktorá aktivuje výstup len keď je vnútorný potenciál neurónu kladný. [9]
$$f(x) = \begin{cases} -1, & \text{ak } x < 0 \\ 1, & \text{ak } x \geq 0 \end{cases}$$



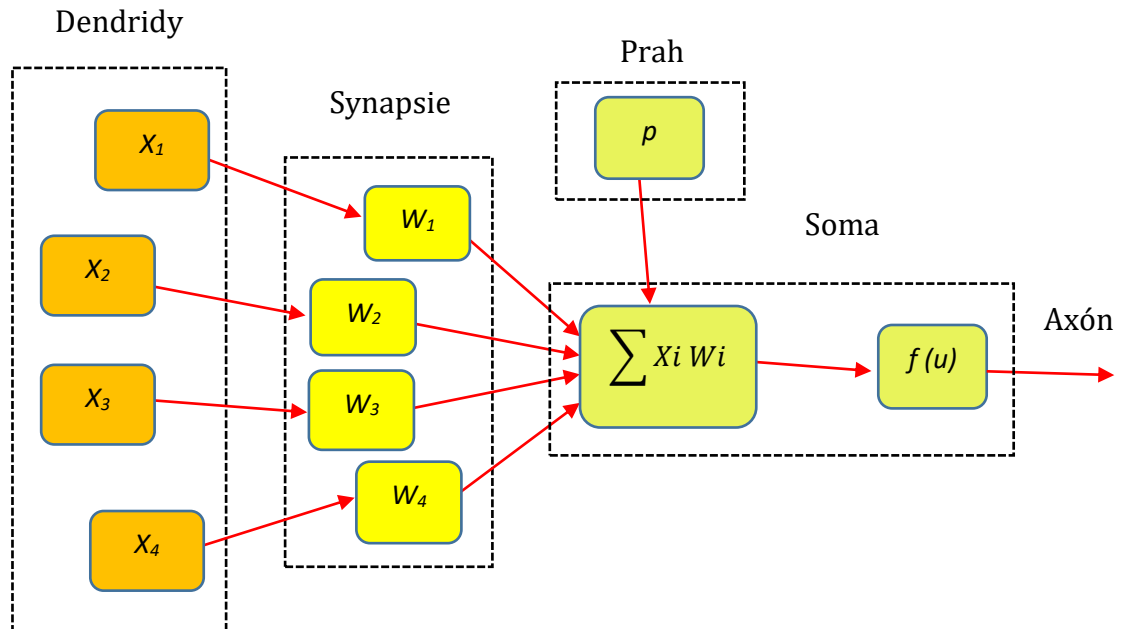
Obr. 2.5 Skoková prenosová funkcia

Tretím, ale nie posledným typom je sigmoidná prenosová funkcia [Obr. 2.6], ktorá je najpoužívanejšia prenosová funkcia, kde λ určuje strmosť funkcie. Pri tejto funkcii existuje spojitá derivácia v každom bode, čo sa využíva v učení neurónovej siete algoritmom spätného šírenia chyby. [9]
$$f(x) = \frac{1}{1+e^{-\lambda x}}$$



Obr. 2.6 Sigmoidná prenosová funkcia

Pre lepšie pochopenie je zobrazený matematický model neurónu popísaný rovnicou (1. 1), ktorý sa používa ako analógia k biologickému neurónu [Obr. 2.7].



Obr. 2.7 Matematický model biologického neurónu

2.3 Neurónová sieť

Neurónové siete pozostávajú z matematických modelov – formálnych neurónov, ktoré sú medzi sebou pospájané rovnakým štýlom ako biologické neuróny synaptickými väzbami s dendridmi ostatných neurónov. Umelá sieť má týchto spojení ešte viac ako biologická. Počet neurónov a spôsob ich prepojení sa nazýva architektúra alebo topológia danej siete. Vstupné a výstupné neuróny tvoria receptory a efektory, sú medzi sebou pospájané, aby vytvárali kanál na šírenie signálu. Vo formálnom modeli sa tieto kanály nazývajú cesty. Šírenie informácií a signálu prostredníctvom nich prebieha ako šírenie stavov jednotlivých neurónov po ceste. Stavby neurónov v sieti, topológia siete a všetky synaptické váhy privedené ostatnými spojeniami tvoria výslednú konfiguráciu neurónovej siete.

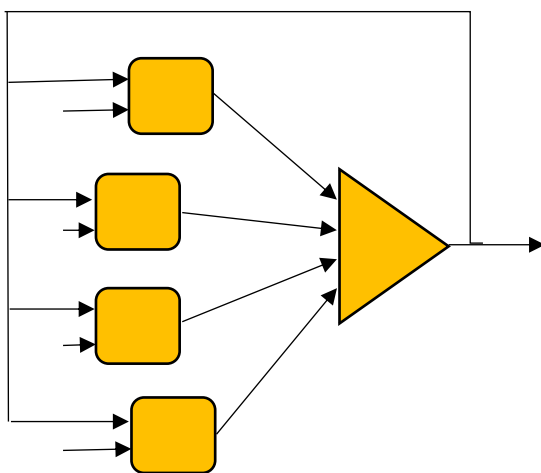
Dôležitou vlastnosťou pri tvorbe topológie siete je rozhodnúť sa, či je potrebné vytvoriť sieť s dopredným šírením signálu alebo spätnoväzobným šírením signálu. Sieť s *dopredným šírením signálu* funguje na takom princípe, že dovolí signálu šíriť sa len jedným smerom, teda od vstupu k výstupu, na ktorom je pripojený ďalší neurón, ktorému je predaná hodnota z výstupu predchádzajúceho neurónu. Nie sú tu spätnoväzobné smyčky, čo znamená že žiadna vrstva nemôže ovplyvňovať sama seba, len pričleňuje určitému vstupu výstup. Najviac rozšírený príklad použitia je rozlišovanie vzorov. Druhým typom je sieť so *spätnoväzobným šírením signálu*. Signál tu síce postupuje od vstupu na výstup, ale v tejto sieti sú povolené spätnoväzobné smyčky. [9] Takto vytvorené siete sú veľmi výkonné, ale extrémne komplikované kvôli ich neustálej dynamike. Siete so spätnoväzobným šírením signálu neustále menia svoj stav, upravujú váhy dokým nedosiahnu rovnovážny stav. V rovnovážnom stave zostávajú do doby pokiaľ sa nezmení vstup a nie je nutné znova zmeniť stav. Siete, ktoré k rovnovážnemu stavu konvergujú, nazývame stabilnými. Ak nastane prípad, že sieť začne divergovať, hovoríme o nestabilnej neurónovej sieti.

Umelá neurónová sieť sa skladá z troch vrstiev. Sú nimi vrstva vstupná, vrstva skrytá a vrstva výstupná. Vstupná vrstva má za úlohu posielat' signály do ďalších vrstiev siete. Skryté vrstvy majú za úlohu spracovávať vstupné údaje na základe váh a spojení medzi jednotlivými skrytými vrstvami. Výstupná vrstva má za úlohu informovať o výstupe zo spracovávaní dát ostatnými vrstvami. Zaujímavým typom je jednoduchá sieť, pretože skryté vrstvy si môžu vytvárať vlastnú reprezentáciu vstupu. Váhy medzi vstupnou a skrytou vrstvou sa určujú v čase, keď sú všetky skryté jednotky aktívne. Takže pri zmene váh si skrytá jednotka, vrstva určuje sama čo získaná váha znamená. [9] [11] Ďalej poznáme jedno-vrstvové siete, kde sú všetky jednotky pospájané vzájomne medzi sebou, čo si vyžaduje veľký výpočetný výkon a tým je náročnejšia na technické vybavenie. Menej náročná na výpočetný výkon je viacvrstvová sieť, kde sú pospájané medzi sebou iba susedné vrstvy, čo znižuje výpočetnú náročnosť.

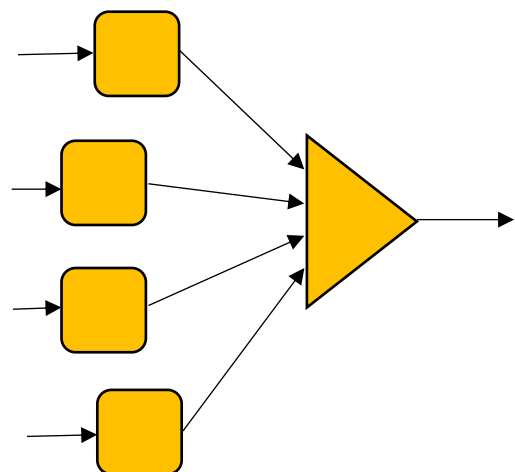
Neurónové siete sa vyvíjajú – tvoria postupne, časom sa vnútorné spojenia spolu so stavmi neurónov, váhami jednotlivých vstupov prispôbujú podľa

aktuálnej potreby. Tieto postupné zmeny sa nazývajú globálnou dynamikou, ktorá sa delí na tri fázy postupných operácií. Prvá fáza sa nazýva *architectural*. Pri tejto fáze sa mení topológia, čiže vzájomné prepojenia, druhá fáza sa nazýva *computational*, prepočítavajú sa jednotlivé váhy a posledná fáza sa nazýva *adaptive*, pri ktorej sa mení konfigurácia neurónovej siete. Tieto fázy však nezodpovedajú biologickému nervovému systému, v ktorom všetky prebiehajú súčasne. Globálnu dynamiku neurónových sietí zvyčajne určujú počiatočné podmienky a požiadavky k úlohe na čo je sieť určená a taktiež matematické rovnice alebo pravidlá, ktoré určujú vývoj topológie, stavov a konfigurácie v reálnom čase. Zmeny sú kontrolované týmito pravidlami a prebiehajú podľa aktuálneho požiadavku. [9]

Pre ilustráciu je zobrazený obrázok siete so spätnoväzobným šírením signálu [Obr. 2.9] a sieť s dopredným šírením signálu [Obr. 2.8].



Obr. 2.9 Sieť so spätnoväzobným šírením signálu



Obr. 2.8 Sieť s dopredným šírením signálu

2.4 Učenie neurónovej siete

Každá neurónová sieť obsahuje informácie o váhach, ktoré sú uložené v matici váh \vec{w} . Učenie je pridelenie váh jednotlivým vstupným vektorom \vec{x} . Pri učení neurónová sieť na základe získaných znalostí musí upravovať maticu váh tak, aby sa minimalizovalo riziko chyby a systém dosahoval čo najpresnejšie výsledky. Podľa

toho či je daná sieť schopná meniť maticu vektorových váh, sa neurónové siete delia na dve skupiny. [12] Prvou sú *fixované siete* – váhy nemôžu byť za žiadnych okolností zmenené ani pri odchýlkach výstupnej funkcie. Znamená to, že $dW/dt=0$. Tento typ neurónových sietí je menej používaný kvôli väčšej chybovosti. Druhým typom sú *adaptívne siete* – váhy môžu byť upravované počas procesu učenia pri zistení odchýlky od očakávanej hodnoty, to znamená, že dW/dt sa nerovná nule.

Učenie neurónových sietí sa rozdeľuje podľa mnohých aspektov. Najdôležitejšia je kategória, podľa ktorej ich delíme na dve veľké časti. Je to učenie *s učiteľom* (supervised learning) a učenie *bez učiteľa* (unsupervised learning).

Najskôr bude popísané učenie *s učiteľom*. Pri tejto metóde učenia musíme vedieť, aký sa očakáva výstup od vstupných hodnôt. Sieť sa v tomto prípade učí priradiť k vstupnému vektoru výstupný vektor z veľkého množstva dát, aby mohla následne reagovať na ľubovoľné dáta. Táto metóda tiež upravuje váhy vektorov na základe počítania chybovej funkcie, aby vstupná hodnota skutočne spĺňala očakávanú hodnotu výstupu (error-correction learning), teda nedochádza ku konvergencii chyby pri spracovaní vstupnej hodnoty neurónovou sieťou. Chyba medzi očakávanou a vypočítanou hodnotou vektoru je minimálna. [12] Najčastejší algoritmus používaný pri procese učenia sa nazýva LMS (Least mean square) convergence, ktorý bol vynájdený na Stenfordskej univerzite v roku 1960 profesorom Bernardom Widrovom a Tedom Hoffom. Je založený na počítaní gradientného poklesu koeficientov, ktoré vykazujú rozdiel medzi očakávanou a skutočnou hodnotou.

Druhý spôsob učenia je učenie *bez učiteľa*. Pri tomto spôsobe učenia nemá sieť znalosti o očakávanej hodnote výstupu. Sieť sa rozhoduje len na základe vstupných dát, detekuje ich spoločné vlastnosti a podľa nich nastavuje jednotlivé váhy. Vo výslednej funkcii sa však môžu objaviť väčšie nepresnosti ako pri učení s učiteľom, keďže sa nepočíta žiadna chybová funkcia a dáta sa pri prechode sieťou nekorigujú až v takej miere ako napríklad pri korekcii pomocou algoritmu Least mean square convergence. [12]

3 KONVOLUČNÉ NEURÓNOVÉ SIETE

3.1 Základné informácie

Konvolučné neurónové siete sú veľmi podobné obyčajným neurónovým sieťam. Sú taktiež zložené s matematických modelov neurónov, ktoré majú premenlivé hodnoty váh a prahy. Každý neurón dostane vstupy, vypočíta skalárne súčiny a sleduje, či netreba upraviť lokálne váhy. Obsahujú jednoducho diferencovateľnú funkciu, ktorá rozloží obrázok na pixely a postupne ich analyzuje až dospeje k výsledku. Počítajú chybovú funkciu a platia tu všetky postupy pri tvorbe architektúry a konfigurácie ako pri neurónových sieťach.

Konvolučná neurónová sieť predpokladá, že vstup je obrázok, z ktorého je možné implementovať isté vlastnosti priamo do architektúry neurónovej siete. Prispeva to k vytvoreniu efektívnejšej funkcie a znižuje počet nutných parametrov siete, tým je sieť značne jednoduchšia. Konvolučná neurónová sieť sa skladá z viacerých konvolučných vrstiev s podvzorkovacími vrstvami, za ktorými je plne prepojená vrstva, ktorá obsahuje variabilné spojenia medzi susednými bunkami. [11]

Hlavným zámerom je identifikácia objektov na základe preddefinovaných vzorov, kde nezáleží na posunutí, deformácii či zmene veľkosti objektu. Analogicky by sa konvolučná neurónová sieť dala prirovnať k postupnému zaostrovaniu objektu zisťovaním detailov. Napríklad prvá vrstva by rozpoznala tmavé a svetlé body, nasledujúca by rozpoznávala hrubé obrysy objektov na obrázku, ďalšia by rozpoznávala tvary, nasledujúca objekty až po vrstvu, ktorá by bola schopná rozlíšiť, že na obrázku sa nachádza automobil.

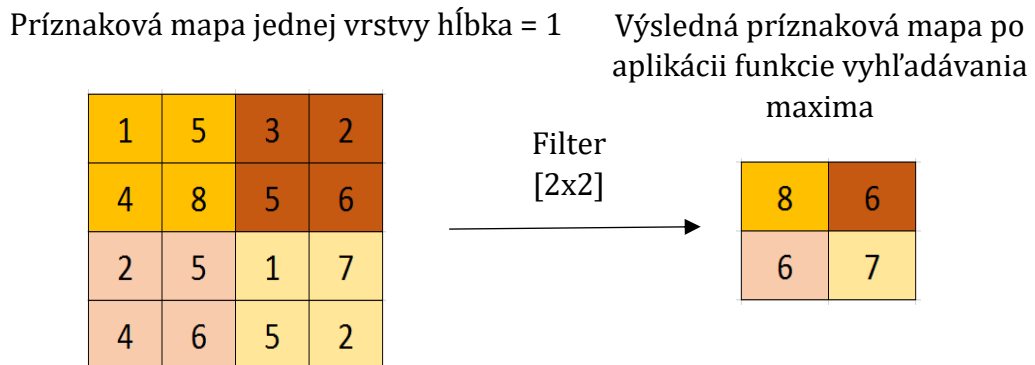
Pri komplexných systémoch zistí, že na aute sa nachádzajú logá sponzorov, trať, diváci na tribúnach a prítomnosť ďalších automobilov, z čoho pochopí kontext obrázku. A ako výstup by sme dostali, že sa jedná o automobilový závod.

3.2 Architektúra konvolučnej neurónovej siete

V tejto časti sú popísané jednotlivé vrstvy siete. Prvá popisovaná vrstva sa nazýva konvolučná vrstva. *Konvolučná vrstva alebo convolutional layer*, pozostáva z filtrov, ktoré majú relatívne malú veľkosť, ale sú schopné rozšíriť sa po celej dĺžke a šírke vstupného objemu dát a učiť sa – získavať príznaky z vstupného obrázku. Pri presúvaní sa posúva filter, nazývaný tiež recepčné pole, po celej šírke a následne výške vstupného obrázku a počíta skalárne súčiny medzi záznamami filtru a vstupmi z obrázku postupne na všetkých pozíciách. Na jeden obrázok sa väčšinou aplikuje viacero recepčných polí s rôznymi hodnotami zdieľaných vektorových váh. Po aplikovaní recepčného poľa na vstupný obrázok sa dostane príznaková mapa. Výsledný počet príznakových máp zodpovedá počtu recepčných polí aplikovaných na obrázok. Ako príklad vstupu zoberieme obrázok o veľkosti $[32 \times 32 \times 3]$ pixely, filter bude mať veľkosť 5×5 pixelov. Každý neurón v konvolučnej vrstve bude mať kapacitu váh pre oblasť $[5 \times 5 \times 3]$ pixelov na vstupnom obrázku, čo znamená celkovo 76 váh - $5 \times 5 \times 3$ plus prah 1. [11] Číslo 3 je hĺbka RGB obrázku, kde farby zaberajú tri kanály. Pri identifikovaní príznaku v konvolučnej vrstve sa snaží zmenšiť príznakové pole buď priemerovaním alebo hľadaním maxima pre efektívnejšie spracovávanie.

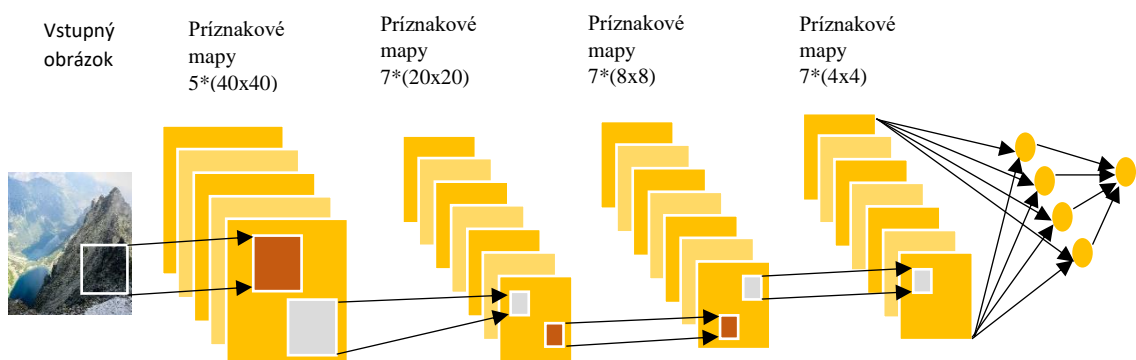
Vrstvy vykonávajúce túto funkciu sa nazývajú *podvzorkovacie vrstvy* alebo *subsampling layers*. Nachádzajú sa medzi dvoma konvolučnými vrstvami. Najbežnejšie používaná funkcia je „*hľadanie maxima*“ alebo *max pooling* [Obr. 3.1]. Táto funkcia progresívne redukuje priestorovú veľkosť a znižuje počet parametrov, s ktorými sieť pracuje, a tým znižuje potrebný výpočetný výkon. Funkcia na hľadanie maxima pracuje nezávisle v každej oblasti aj na každom bode hĺbky danej oblasti čo znamená, že pri vyššie spomenutom príklade oblasti $[5 \times 5 \times 3]$ pixelov bude priestorovo zmenšovať 3 príznakové mapy použitím hľadania maxima z bodov matice príznakov. Najčastejšie sa používa hľadanie maxima zo štyroch prvkov $[2 \times 2]$ pixely, čo zredukuje veľkosť vrstvy o 75%. [13] Hľadanie maxima nie je učenie neurónovej siete v pravom slova zmysle. Namiesto učenia sa redukuje veľkosť tak, že bloky o veľkosti $[n \times n]$ sa minimalizujú na jednu hodnotu. Ak práve

táto hodnota obsahuje chybu z predošlej vrstvy tak sa posiela späť na vrstvu kde vznikla.



Obr. 3.1 Grafické znázornenie max-pooling algoritmu

Tento proces prebehne viac krát podľa potreby a nakoniec je výstup predaný na vstup *plne prepojenej neurónovej siete* alebo *fully-connected neuron layer*, ktorá sa nachádza na konci za všetkými konvolučnými aj podvzorkovacími vrstvami. Tu sa všetky výstupy spájajú do jedného neurónu, ktorý má len jeden rozmer teda konvolučná vrstva sa k nemu už nemôže pripojiť. Výstupy z neurónov sa pretransformujú do jednoduchého vektoru. Prakticky nemôžeme vytvoriť dvojrozmernú alebo trojrozmernú konvolučnú sieť, pretože príznakové mapy sú jednorozmerné matice. [13] Viacrozmernosť sietí vzniká hĺbkou vstupného obrazu. Napríklad RGB model má tri kanály na farbu. Táto hĺbka sa však spracúva po jednotlivých vrstvách. Príklad konvolučnej neurónovej siete je na obrázku [Obr. 3.2].



Obr. 3.2 Príklad konvolučnej neurónovej siete s jednotlivými vrstvami

3.3 Učenie konvolučnej siete

Pre učenie konvolučnej siete sa používa najčastejšie algoritmus spätného šírenia, ktorý aplikujeme na konvolučnú sieť. Ako bolo vyššie spomenuté daná sieť sa skladá z konvolučných, podvzorkovacích vrstiev, ktoré sa striedajú a plne prepojených vrstiev na konci siete. Metódu spätného šírenia chyby môžeme použiť pre plne prepojenú vrstvu. Pre potreby zvyšných dvoch sa algoritmus nasledovne mierne upravuje.

Za konvolučnou vrstvou l sa nachádza podvzorkovacia vrstva $l+1$. Aby sa mohli spočítať chyby v jednej vrstve musia sa spočítať chyby cez všetky prvky v nasledujúcej vrstve. Je to kvôli použitiu metódy na zmenšovanie vstupov napríklad max-pooling algoritmu, kde sa vyberá najväčší prvok z matice, ktorý je predávaný vrstve ako vstup. To znamená že jeden pixel podvzorkovacej vrstvy zodpovedá viacerým pixelom konvolučnej vrstvy. Konvolučné neurónové siete sa ľahšie naučia kvôli zdieľaniu váh, čo zapríčiňuje menšiu početnosť parametrov, ktoré sa musí učiť. Tieto siete taktiež pracujú priamo s obrazom, bez pretransformovania do rôznych vektorov a kompresie ich veľkosti. Tolerujú posuny alebo deformácie objektov na obrázkoch a dokážu spracovať väčšie obrazy za rovnaký čas. [14]

3.4 Oblasti používania konvolučných neurónových sietí

Jednou z najširších oblastí použitia konvolučných neurónových sietí je jednoznačne identifikácia objektov, či rozpoznávanie tvárí. Zdokonaľovaním v tomto smere sa najviac zaoberajú firmy ako Google, Facebook či Microsoft, pretože identifikácia objektov v obraze je pre nich dôležitá kvôli používaniu tejto funkcie vo viacerých produktoch, ktoré vyvíjajú.

Z pohľadu identifikácie jednotlivých objektov z obrázkov je síce človek stále lepší, ale neplatí to pre všetky úlohy. Napríklad pri rozpoznávaní značiek, čo zaujíma hlavne tímy pracujúce na vývoji samočinných automobilov, kde sa hľadá

technológia ako bezpečne riadiť tieto autá bez rizika zlého vyhodnotenia situácie, dosiahla umelá inteligencia menšiu chybovosť ako človek.

Vo vysokej miere sa využívajú aj pri detekcii mitóz, odlišností od iných tkanív, pozorovaných mikroskopmi. Odlišnosti tkanív a ich spozorovanie sú kľúčové pre včasnú diagnózu niektorých chorôb ako napríklad rakoviny. [11]

Rozoznávanie rôznych miest na svete nerobí ľuďom až taký problém. Keď na obrázku nenájde všeobecne známy objekt typický pre nejakú oblasť ako napríklad Sochu slobody, Eiffelovú vežu, Grand Canyon, podľa ktorého by sme miesto automaticky identifikovali, tak si začneme všímať okolie, prítomnosť reklám, nápisov, áut, rás alebo iných znakov, podľa ktorých sa snažíme dospieť k identifikácii miesta. Toto dokáže aj sieť s názvom PlaNet. Je natrénovaná pomocou obrovskej databázy obrázkov z Google street view a dokáže behom chvíle odhadnúť polohu fotky ľubovoľného miesta na planéte. Chybovosť tohto testu je síce vyššia. Ulicu neuhádne s pravdepodobnosťou 96,4%, mesto 90%, štát 71,6% a kontinent 52%. [15] Tieto výsledky sa na prvý pohľad môžu zdať zlé, ale v porovnaní s človekom je sieť PlaNet presnejšia, teda je v nadľudskej úrovni.

Vrátenie farieb do čiernobielych obrazov je oblasť, v ktorej tiež dosahuje dobré výsledky. Dát na tréning je nespočetné množstvo, keďže z každého farebného obrázku vieme vyrobiť jednoduchý čiernobiely obrázok. [16] Podobným príkladom je vyextrahovanie štýlu umeleckého diela. Štýl tvorby maliarov dokážeme bez problémov rozoznať. Obrazy jedného umelca, napríklad obrazy Pabla Picassa, sa vyznačujú ostrými tvarmi, jasnými farbami a inými typickými znakmi. Konvolučná neurónová sieť sa dokáže tieto znaky naučiť, získať relevantné informácie o štýle jeho tvorby a upraviť ľubovoľnú fotografiu do naučeného štýlu. Vlastné fotky do veľkosti maximálne 500x500 pixelov si môžeme takýmto spôsobom dať upraviť na adrese <https://deepart.io/> [25], kde je možné vybrať si z prednastavených štýlov alebo vložiť vlastný obraz, podľa ktorého sa má sieť naučiť štýl maliara a následne ho aplikovať na obrázok.

Zaujímavou oblasťou je aj hranie hier s množstvom kombinácií a ťahov náročnejších ako šach. Pri testovaní bola umelá inteligencia v nadpolovičnej väčšine

lepšia ako profesionálny hráči, avšak sieť si vo viacerých prípadoch počínala ťahmi, ktoré by človek nespravil. Ak mala napríklad 60% šancu na výhru v štyroch ťahoch, zvolila si možnosť s 98% šancou na výhru s tridsiatimi piatimi ťahmi. Taktiež používala kroky, ktoré sa najskôr zdali chybné, no vo výsledku sa ukázali ako rozumné. Pri zistení, že prehrá, začala skúšať zvláštne ťahy, na ktoré by súper chybne zareagoval a tým sa pripravil o výhru. Program, ktorý porazil svetového šampióna v hre GO 5:0, Lee Sedola, bol vyvinutý spoločnosťou Google. [17]

Popísané neboli samozrejme všetky oblasti, kde sa konvolučné siete používajú alebo práve vyvíjajú, pretože ich je veľmi veľké množstvo. Keďže sa táto oblasť dynamicky rozvíja hľadajú sa ďalšie možnosti použitia konvolučných neurónových sietí v rôznych oblastiach.

4 INŠTALÁCIA PRE PLATFORMU WINDOWS

4.1 Dostupné frameworky

Pred samotnou inštaláciou bolo nutné zvoliť vhodný framework, ktorý je často využívaný, má dostatočnú podporu a dokumentáciu. Dôležité je, či sa daný framework vyvíja a zlepšuje alebo sa z dôvodu jeho neefektívnosti prestáva používať. V tejto kapitole sú popísané rôzne dostupné frameworky, z ktorých je následne vybraný jeden, ktorý bude možné implementovať do operačného systému Windows. Popis je zložený s oficiálneho popisu priamo zo stránok Nvidia developers a druhá časť je vybraná z recenzií. Popis jednotlivých frameworkov je čerpaný z [18] [19]. Pre prehľadnosť sú zdroje uvedené na začiatku kapitoly.

TensorFlow je open-source framework, ktorý používa numerické výpočty s použitím grafov dátového toku. Vytvorený bol spoločnosťou Google, ktorá sa výskumom v danej oblasti zaoberá v laboratóriách Umelej Inteligencie. Framework podporuje rozhrania C++ a Python. Z recenzií som zistil, že pri implementovaní modelu je nutné napísať množstvo kódu. Vývojári v súčasnosti vydali novú knižnicu, ktorá by prácu mala zjednodušiť a tým prilákať širší počet užívateľov.

Theano je jedným z prvých frameworkov, ktoré začali pracovať z konvolučnými neurónovými sieťami. Efektívne definuje matematické operácie a výrazy. Zahŕňa tiež prácu s viacrozmernými reťazcami. Hĺbkové učenie sa realizuje pomocou Python rozhrania spojeného s Numpy balíčkom pre Python, čo je základný balíček pre vedecké výpočty. Nevýhodou je, že nepodporuje paralelne viac grafických kariet, čo znižuje dosiahnuteľný výkon.

Keras je modulárny a minimalistický nástroj. Celá knižnica je napísaná v Pythone. Vývojári sa zamerali hlavne na možnosť rýchleho aplikovania na konkrétne dáta a jednoduchosť. Podporované rozhranie je Python. Jazyk je jednoduchý a tiež dobre zdokumentovaný. Keras pracuje ako nadstavba - rozhranie do Theano alebo TensorFlow kvôli zjednodušeniu práce so spomínanými frameworkami.

Caffe je framework pre hĺbkové učenie, ktorý bol pri vývoji zameraný na rýchlosť a modularitu. Vyvíjaný bol na Univerzite v Berkley. Podporuje rozhrania C, C++, Python, MATLAB a rozhranie príkazového riadku. Zameraný je na počítačové videnie. Je asi päť krát rýchlejší ako Keras. V súčasnosti je veľmi rozšírený, robustný a rýchly. Inštalácia je náročnejšia ako v predchádzajúcich prípadoch.

The Microsoft Cognitive Toolkit, predtým známy ako CNTK, vznikol zjednotením nástrojov vyvíjaných firmou Microsoft. Cieľom je jednoduché učenie a tréning modelov. Podporuje viacero grafických kariet paralelne a tiež môže bežať na serveroch. Použiteľný je pre reč, text, obrázky a dáta. Podporuje rozhrania Python, C++ a C#. Tento framework nie je až tak populárny. Na internete sa nenachádza také množstvo diskusií a modelov ako napríklad pri Caffe, takže práca s The Microsoft Cognitive Toolkit je ťažšia. Úspechom frameworku je zdolanie svetového rekordu v rozpoznávaní reči, pri ktorom sieť dosiahla ľudskú úroveň.

Torch je framework, ktorý podporuje širokú škálu algoritmov pre hĺbkové učenie a vedecké výpočty. Podporuje rozhrania C, C++ a Lua. Jazyk Lua môže byť výhodou aj nevýhodou, pretože prístup pomocou jazyku Lua nepoužíva žiaden z uvedených frameworkov. Pred používaním Torch je najsôr nutné naučiť sa tento programovací jazyk a až potom používať Torch. Práve kvôli tomuto dôvodu nie je až tak rozšírený.

4.2 Výber Frameworku

Pri mnohých frameworkoch sa inštalácia na operačný systém Windows nedá zrealizovať. Jedinou možnosťou je inštalácia prostredníctvom virtuálneho prostredia Linux. Pri tomto spôsobe však môže vzniknúť problém pri komunikácii virtuálneho stroja s grafickou kartou v počítači. Zbernica PCI – Express môže pri komunikácii medzi fyzickým nainštalovaným operačným systémom a virtuálnym operačným systémom prenášať signál na menších rýchlostiach alebo nemusí pracovať vôbec. Preto sa táto možnosť veľmi nepoužíva, avšak je to individuálne a závisí to od konkrétnej distribúcie Linuxu použitého pre virtualizáciu. Napríklad pre framework Torch sa odporúča použiť distribúciu Ubuntu. Tak isto aj vytvorené aplikácie pracujú vo väčšine prípadov pod Linuxovými systémami.

Pre bakalársku prácu bol vybraný Caffé framework. Caffé framework nemá oficiálnu podporu pre systémy Windows. Je to open source framework. Rozširuje sa medzi množstvo ľudí a vznikajú komunity vývojárov či užívateľov, ktorých daná problematika zaujíma. Takéto skupiny diskutujú o možnostiach implementácie do operačného systému Windows, o tvorbe modelov pre učenie a trénovanie konvolučných neurónových sietí, o problémoch pri tvorbe vlastných modelov a rôznych iných veciach týkajúcich sa Caffé frameworku. Caffé framework bol vybraný aj kvôli možnosti overenia funkčnosti úspešným skompilovaním bez hlásení „errors“.

Caffé je jeden z najpoužívanejších a najrýchlejších frameworkov. Bol vyvinutý na Univerzite v Berkley v laboratóriách umelej inteligencie. Na jeho zdokonalení majú veľkú zásluhu aj rôzni iní programátori, ktorí počas prvého roku od vydania upravovali zdrojový kód a hľadali chyby. Základom bol projekt Yangqing Jia v rámci doktorandského štúdia na Univerzite v Berkley. Veľkou prednosťou Caffé frameworku je jeho rýchlosť. [20] Pri súťaži dokázalo spracovať 60M obrázkov za deň na grafickej karte Nvidia K40. Hardvér sa neustále vylepšuje, čo môže prispieť k väčším rýchlostiam v budúcnosti.

4.3 Úvod k inštalácii

Ako bolo spomenuté v kapitole 4.2 *Výber frameworku*, väčšina frameworkov, ktoré sú zamerané na neurónové siete – konvolučné alebo bežné pracuje pod operačným systémom Linux. Najviac používané sú v distribúcii Ubuntu. Pre systémy Windows vývojári neposkytujú oficiálnu podporu. V bežnom Linuxovom operačnom systéme prebieha inštalácia v niekoľkých jednoduchých krokoch pomocou príkazového riadku. Návod sa nachádza na oficiálnych vývojárskych stránkach konkrétneho frameworku.

Pred implementáciou Caffé frameworku na operačný systém Windows bolo nutné prečítať množstvo diskusií na rôznych fórach a hľadať možnosti, akým spôsobom splniť túto úlohu. Najlepším riešením bola implementácia Caffé

frameworku do vývojového prostredia Microsoft Visual Studio 2013. Pri inštalácii vzniklo množstvo problémov, ktoré bolo nutné vyriešiť. Pred samotnou inštaláciou bolo nutné pripraviť rôzne verzie knižníc, platformy CUDA a tiež Microsoft Visual Studia. Na testovanie bolo použité Microsoft Visual Studio 2012, 2013, 2015. Jediné vývojové prostredie, kde sa implementácia podarila bez chýb je verzia 2013. Tiež bolo nutné vyskúšať rôzne verzie platformy CUDA a knižníc cuDNN od Nvidia vo verziách v3, v4 a v5. Nakoniec je použitá verzia v4, s ktorou Caffé framework spolupracuje. Tiež najnovšia verzia NuGet Packages v3 nie je vhodná kvôli následným problémom s kompiláciou.

4.3.1 Open Source licencia

Jednotlivé frameworky sú skoro vo všetkých prípadoch pod open source licenciou. Open source znamená, že software môže byť bezplatne šírený a upravovaný za určitých podmienok. Open source projekty sú projekty, ktoré prijali otvorenú výmenu názorov vývojárov s užívateľmi. Znamená to, že prijali účasť na spolupráci s ľubovoľnými ľuďmi so záujmom o danú problematiku. Open Source prekladáme ako „otvorený zdrojový kód“, no v skutočnosti sa nejedná len o voľne prístupný zdrojový kód. Distribučné podmienky musia spĺňať určité kritéria. Patrí sem bezplatná redistribúcia, zdrojový kód, odvodené práce, integrita autorovho zdrojového kódu, nesmú diskriminovať jednotlivcov alebo skupiny ani slúžiť na účely diskriminácie, musia distribuovať licenciu, licencia nesmie obmedzovať iný softvér, musí byť technologicky neutrálna a rovnaká pre celý produkt. Softvér, v našom prípade framework, pod touto licenciou je „neobmedzený“. Môžeme ho používať neupravený v pôvodnej forme alebo ak má niekto nápad ako zlepšiť daný produkt tak ho môže upraviť, optimalizovať, vylepšiť, zefektívniť a takto upravený zdieľať s ostatnými užívateľmi. Každá úprava musí byť riadne zdokumentovaná a kód zakomentovaný kvôli prehľadnosti pre ostatných užívateľov. [21]

Pravidlá dokumentácie si určuje každý vývojársky tím Open Source platformy osobitne a nájdeme ich medzi základnými informáciami na ich stránkach. Myšlienka sprístupniť zdrojový kód sa ukázala ako veľmi prospešná tým, že kód nekontroluje len úzka skupina vývojárov, ale nepomerne väčšie množstvo užívateľov. Chyby

v pôvodnom kóde sa nájdu a opravia skôr, prípadne sa navrhnu lepšie spôsoby ako implementovať daný problém. Open source licencia prispieva k rýchlemu napredovaniu v oblasti vývoja softvéru a tak isto šetrí finančnú náročnosť daných projektov.

4.4 Inštalácia po jednotlivých krokoch

4.4.1 Potrebný softvér

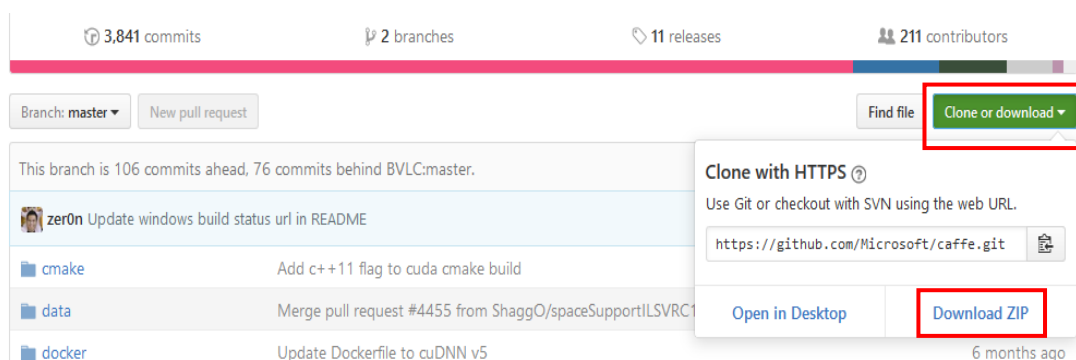
Vybraný Caffé framework sa bude spúšťať pomocou vývojového prostredia Microsoft Visual Studio. Vývojové prostredie Microsoft Visual Studio bolo vybrané aj kvôli tomu, že podporuje prácu s prídavnými knižnicami CuDNN od Nvidia a spolupracuje s platformou CUDA, ktorá značne zrýchľuje výpočty. Dôležitá je verzia Microsoft Visual Studia. Mohlo by sa zdať, že najnovšia verzia je najlepšia, no opak je pravdou. Caffé framework je možné používať pod operačným systémom Windows len vo vývojovom prostredí Microsoft Visual Studio 2013. Testované boli aj verzie 2010, 2012, 2015, ale inštalácia sa v týchto vývojových prostrediach nepodarila. Tiež je potrebný program Notepad++ [22], namiesto neho je možné použiť aj textový editor nainštalovaný v operačnom systéme Windows, ale Notepad++ je prehľadnejší a ľahšie sa nájdu riadky kódu, ktoré je nutné upravovať. Ďalej je potrebná grafická karta od Nvidie, ktorá podporuje platformu CUDA pre výkonnejšiu verziu s použitím platformy CUDA. Zoznam grafických kariet a bližšie špecifikácie aktuálnej nainštalovanej grafickej karty v počítači sú na stránke Nvidia [23]. Pre inštaláciu je potrebné mať 64 bitovú verziu Visual Studia. Z toho vyplýva, že je nutné disponovať 64 bitovým operačným systémom Windows.

Pri inštalácii je dôležité dodržiavať popísané verzie jednotlivých súčastí, pretože dané frameworky, ktoré pracujú s konvolučnými neurónovými sieťami sú primárne určené pre Linux a ich implementácia na operačný systém Windows je náročná. Zdrojový kód Caffé je open source platformou a je značne rozsiahly. Kompilácia zaberie určitý čas. Pri kompilácii je procesor prakticky stále zatážený na 100%. Časť priebehu kompilácie a zobrazenie vyťaženia procesora je vidieť aj v krátkom video návode pre dokončenie inštalácie.

4.4.2 Postup inštalácie s použitím platformy CUDA

1. Prvým krokom je inštalácia Microsoft Visual Studio, ako bolo spomenuté vyššie. Musí sa nainštalovať verzia Visual Studio 2013, pri verzii 2015 sa nedá spustiť viacero knižníc a pri 2010 a 2012 nefunguje libcaffe.h a caffe.h, ktoré sú základom inštalovaného frameworku. Bez nich Caffe framework nie je funkčný.

2. Caffe framework sa dá stiahnuť na adrese <https://github.com/Microsoft/caffe>. Je to stránka, kde diskutujú vývojári, ktorí pracujú na Open Source technológiách, navrhujú svoje riešenia a pomáhajú si riešiť vzniknuté problémy pri rôznych softvéroch či zdrojových kódach. Po prejení na spomínanú stránku sa klikne na zelené tlačidlo *Clone or download*. Po rozbalení ponuky je nutné Caffe framework uložiť kliknutím na „Download ZIP“. [Obr. 4.1]

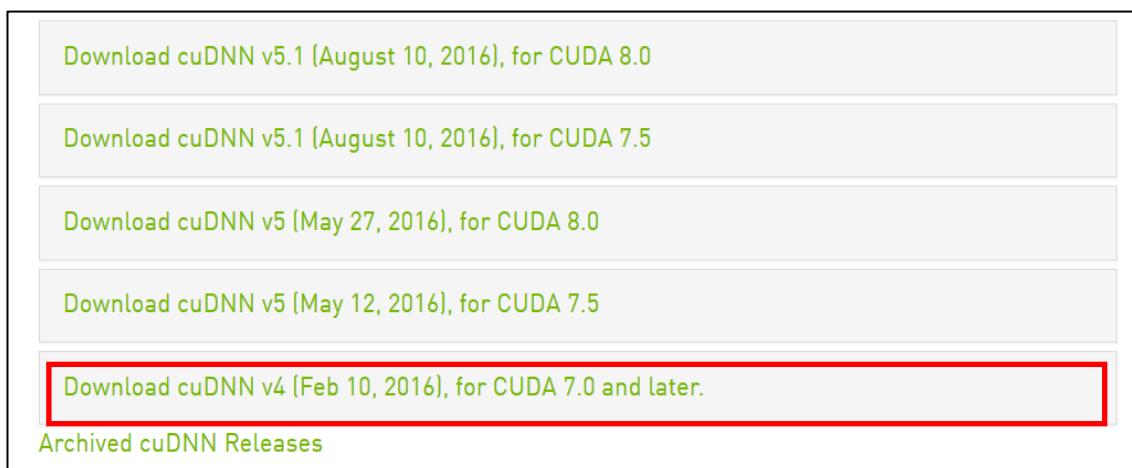


Obr. 4.1 Stiahnutie Caffe frameworku

3. Aby Caffe framework fungoval, musí sa stiahnuť súbor CuDNN od Nvidie. CuDNN (*NVIDIA CUDA® Deep Neural Network library*) je knižnica vyvíjaná Nvidiou pre hĺbkové učenie konvolučných neurónových sietí. Jej funkcie sa spúšťajú pomocou grafického procesoru a slúžia na urýchlenie výpočtov. Knižnica obsahuje implementácie štandardných operácií ako dopredné šírenie signálu, spätné šírenie signálu, normalizačné, združovacie-pooling funkcie, sú tu definované aktivačné vrstvy a mnohé iné. CuDNN je súčasťou NVIDIA Deep Learning Software Development Kit. V knižnici sú optimalizované metódy s odstránenými chybami. Vývojári aplikácií nemusia tráviť čas s ich tvorbou a môžu sa venovať praktickému

využití, trénování a tvorbě aplikací nebo jiného softwaru, který Caffe framework využívá pro svou činnost. CuDNN se používá na hlubkové učení v mnohých frameworkech jako sú Caffe, TensorFlow, Theano, Torch, CNTK.

CuDNN se dá stáhnout na stránce <https://developer.nvidia.com/cudnn>. Následně je nutné zaregistrovat se, vyplnit krátký dotazník, načež budou knihovny využívány a odkliknout souhlas s podmínkami. Po potvrzení o souhlase s podmínkami se zobrazí verze cuDNN. [Obr. 4.2] Opět není nutné stahovat nejnovější verzi knihovny 5.1 pro platformu CUDA 8.0 ale verzi cuDNN v4 (Feb 10 2016) pro platformu CUDA 7.0 a novší. Stahované knihovny třeba rozbalit a přesunout na libovolné místo na disku.



Obr. 4.2 Ukážka stránky s rôznymi verziami CUDY

4. Pre spustenie je tiež veľmi dôležitá platforma CUDA, bez ktorej Caffe framework nebude fungovať. Stáhnúť sa dá priamo zo stránok Nvidia <https://developer.nvidia.com/cuda-downloads>. Potrebne je vybrať aktuálny operačný systém Windows 7 alebo Windows 10 a nainštalovať platformu CUDA. Pre Windows 10 je možné nainštalovať iba najnovšiu verziu, ktorá tiež spolupracuje s cuDNN v4.

5. ZIP súbor stiahnutý z github.com v druhom kroku je potrebné rozbalit do ľubovoľnej zložky v počítači. Zložka má niekoľko desiatok megabajtov. Pre dokončenie inštalácie bude potrebných niekoľko gigabajtov. Z tohto dôvodu si treba vyhraďiť minimálne štyri gigabajty voľného priestoru na disku.

6. V zložke s rozbaleným súborom *caffe-master.zip* je nutné prejsť k súboru *caffe-master/windows/CommonSettings.props.example*, spraviť jeho kópiu a ponechať ju v tej istej zložke.

7. Novo vytvorený súbor *CommonSettings.props* – *kópia.example* treba premenovať na *CommonSettings.props*, potvrdiť zmenu prípony súboru, ktorého typ sa zmení z *example* na *Project Property File*, tiež zmení ikonu na symbol kľúča. Tento súbor je lepšie otvárať v textovom editore Notepad++, ktorý prehľadne naformátuje daný súbor, aby sa dal efektívne upravovať.

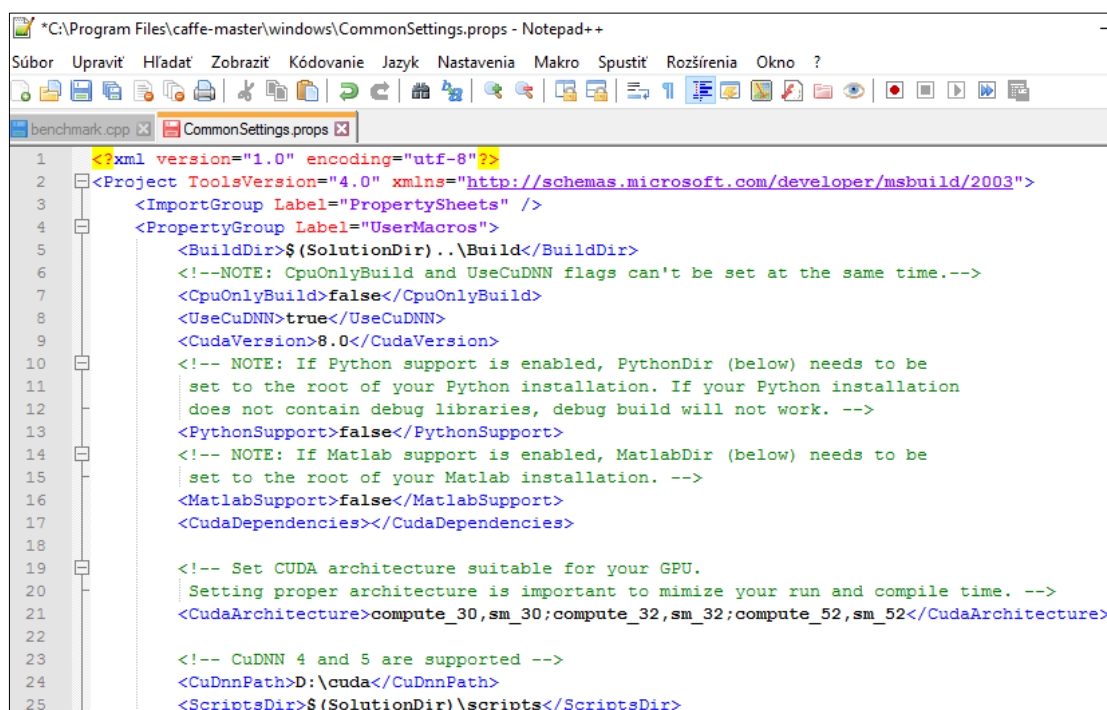
8. V tomto kroku sa musia upravovať niektoré riadky v novo vytvorenom súbore *CommonSettings.props*. Potrebne je skontrolovať, či je správne nastavené *<CpuOnlyBuild>* *<CpuOnlyBuild>* na "false", *<UseCuDNN>* *<UseCuDNN>* na "true", *<PythonSupport>* *<PythonSupport>* a *<MatlabSupport>* *<MatlabSupport>* na "false" kvôli tomu, že nepoužívame ani Matlab ani Python. Pri finálnej kompilácii bude Visual Studio 2013 dva projekty pre Python a Matlab preskakovať. Ak hodnoty nesedia s vyššie uvedenými, tak ich je potrebné zmeniť na uvedené.

Upravovať sa bude ďalej riadok číslo 9 s textom *<CudaVersion>* *<CudaVersion>*, do ktorého sa dopisuje verzia aktuálne nainštalovanej platformy CUDA. V prípade operačného systému Windows 10 verzia 8.0. Staršia verzia 7.0 je podporovaná len Windowsom 7 alebo starším.

Následne sa upravuje riadok číslo 21 s textom *<CudaArchitecture>* *<CudaArchitecture>*. Do tohto riadku sa píše výpočetný výkon aktuálnej grafickej karty (compute capatipility), ktorý sa dá nájsť v technickej špecifikácii grafickej karty na <https://developer.nvidia.com/cuda-gpus>. Tento parameter je dôležitý pre optimalizáciu behu programu a kompilácie, skracať čas potrebný na učenie a tréovanie konvolučných neurónových sietí.

Tretím riadkom, ktorý sa bude upravovať je riadok číslo 24 *<CudaPath>* *<CudaPath>*, kde sa nastavuje cesta ku rozbalenej knižnici cuDNN stiahnutej v treťom kroku. Uvádza sa cesta ku zložke s cuDNN knižnicami. Po rozbalení je to zložka s názvom „cuda“. Ak sa skopíruje napríklad do zložky „caffe“ na disku „D://“, bude zadávaná cesta „ D://caffe“ inak dané knižnice Visual Studio nenájde. Takto

upravený súbor je potrebné uložiť. Ako príklad je zobrazený finálne upravený súbor *CommonSettings.props* . [Obr. 4.3]



```
<?xml version="1.0" encoding="utf-8"?>
<Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
  <ImportGroup Label="PropertySheets" />
  <PropertyGroup Label="UserMacros">
    <BuildDir>$(SolutionDir)..\Build\BuildDir>
    <!--NOTE: CpuOnlyBuild and UseCuDNN flags can't be set at the same time.-->
    <CpuOnlyBuild>false</CpuOnlyBuild>
    <UseCuDNN>true</UseCuDNN>
    <CudaVersion>8.0</CudaVersion>
    <!-- NOTE: If Python support is enabled, PythonDir (below) needs to be
    set to the root of your Python installation. If your Python installation
    does not contain debug libraries, debug build will not work. -->
    <PythonSupport>false</PythonSupport>
    <!-- NOTE: If Matlab support is enabled, MatlabDir (below) needs to be
    set to the root of your Matlab installation. -->
    <MatlabSupport>false</MatlabSupport>
    <CudaDependencies></CudaDependencies>

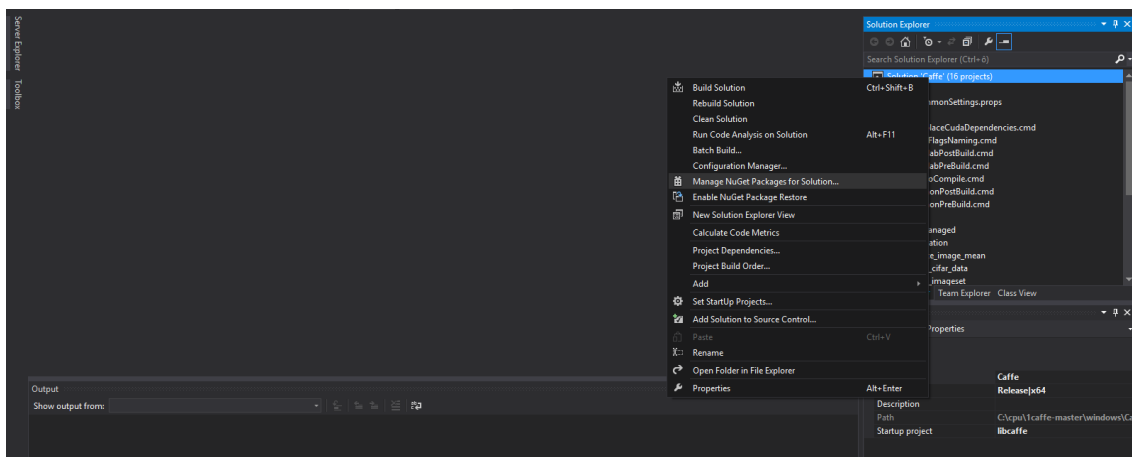
    <!-- Set CUDA architecture suitable for your GPU.
    Setting proper architecture is important to minimize your run and compile time. -->
    <CudaArchitecture>compute_30,sm_30;compute_32,sm_32;compute_52,sm_52</CudaArchitecture>

    <!-- CuDNN 4 and 5 are supported -->
    <CuDnnPath>D:\cuda\CuDnnPath>
    <ScriptsDir>$(SolutionDir)\scripts</ScriptsDir>
```

Obr. 4.3 Ukážka upraveného súboru *CommonSettings.props*

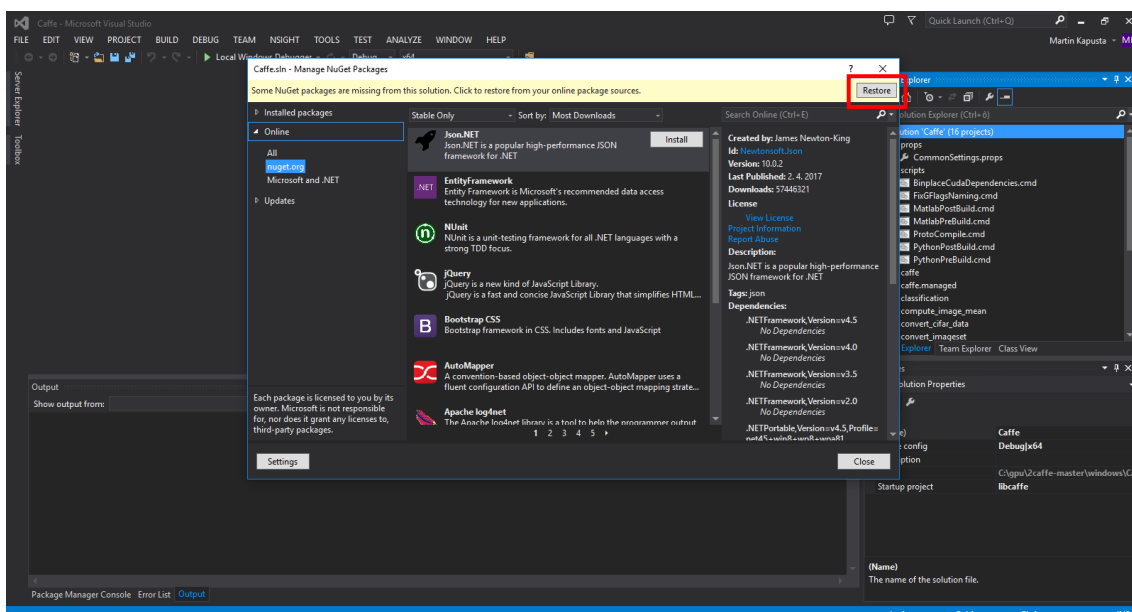
9. V zložke „windows“ v priečinku „caffe-master“ je potrebné otvoriť *Caffe.sln*. Otvára sa pravým tlačidlom → otvoriť v programe Visual Studio 2013. Môže sa objaviť výzva, či chceme otvoriť projekt z nedôveryhodného zdroja. Výzva sa musí potvrdiť a počkať, kým sa otvorí a načíta všetkých 16 projektov. Táto operácia trvá niekoľko minút pre veľký rozsah Caffe frameworku.

10. Pravým tlačidlom myši je nutné kliknúť na *Solution 'Caffe' (16 Projects)* → *Manage NuGet Packages for Solution...* Zobrazené na obrázku [Obr. 4.4].



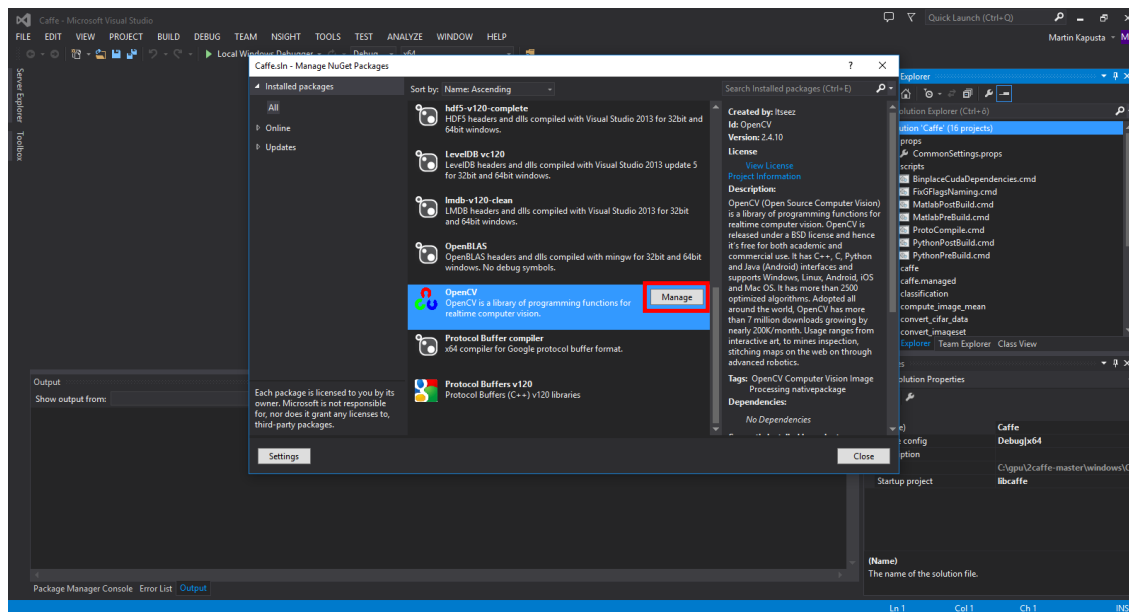
Obr. 4.4 Otvorenie „Manage NuGet Packages for solution...”

11. Otvorí sa okno s názvom *Caffe.sln- Manage NuGet Packages*. V hornej časti tohto okna, v žltom rámečku je napísané „Some NuGet Packages are missing from this solution. Click to restore from your online package sources“. Na konci žltého rámečka je tlačidlo „Restore“, na ktoré je potrebné kliknúť. [Obr. 4.5] Automaticky sa stiahnu všetky potrebné NuGet balíčky. Tento proces trvá niekoľko minút až desiatok minút podľa rýchlosti internetového pripojenia. Visual Studio prestane reagovať, takže počas sťahovania NuGet Packages netreba používať iné funkcie Visual Studio. Po nainštalovaní sa v adresári, v ktorom sa nachádza projekt Caffe, vytvorí zložka „Nuget packages“ s nainštalovanými súborami.

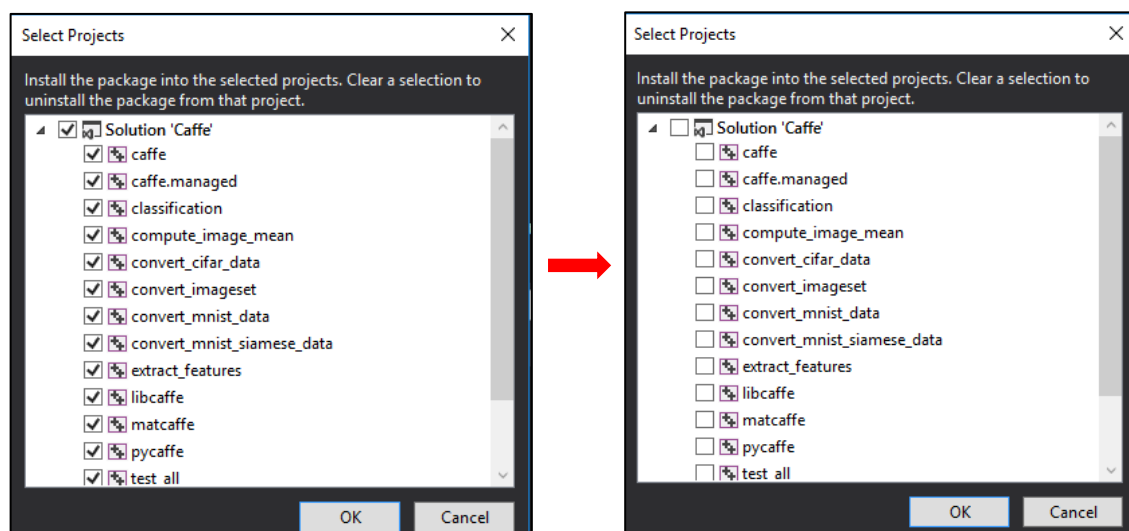


Obr. 4.5 Tlačidlo „Restore“

12. Po dokončení predchádzajúceho kroku sa musí rozkliknúť na ľavej strane okna *Caffe.sln – Manage NuGet Packages*, prvý riadok z menu *Installed packages* → *all*. V zozname nainštalovaných balíčkov treba nájsť knižnicu OpenCV, kliknúť na *Manage*. [Obr. 4.6] Zobrazia sa všetky knižnice z projektu a každá je označená. Pre správnu funkčnosť frameworku sa musia všetky odznačiť a potvrdiť tlačidlom *OK*. [Obr. 4.7] Týmto krokom prebehne odstránenie OpenCV z projektu a môže sa pokračovať ďalej v inštalácii.

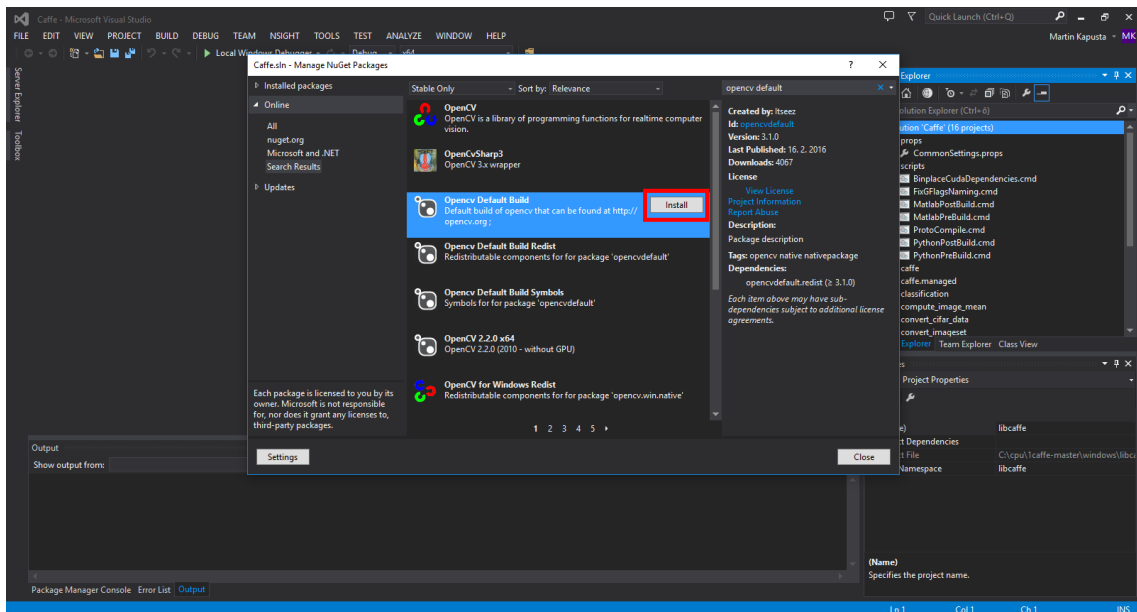


Obr. 4.6 Odinštalovanie pôvodného OpenCV

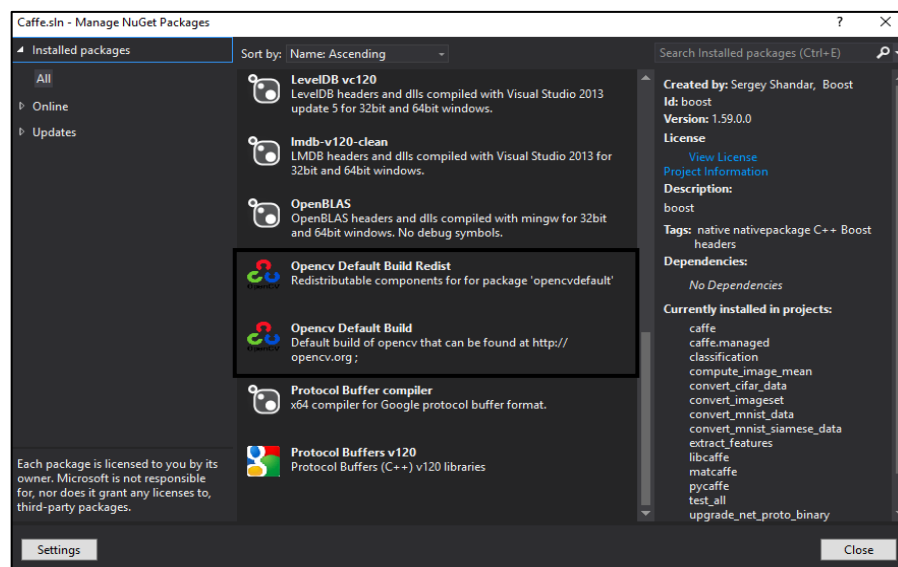


Obr. 4.7 Ukážka odznačenia projektov

13. V ľavom menu sa treba presunúť z *Installed packages* do druhej časti *Online*, v pravom hornom rohu, do poľa vyhľadávanie napísať „opencv default“ a počkať, kým sa automaticky vyhľadajú potrebné balíčky. Následne vybrať OpenCV Default Build a kliknúť na tlačidlo „Install“. [Obr. 4.8] Musí sa označiť všetko v Solution ‘Caffe’ ako pri odinštalácii v predchádzajúcom kroku a potvrdiť OK. Týmto krokom sa nainštaluje OpenCV Default Build a OpenCV Default Build Redist. Výsledok je zobrazený na obrázku [Obr. 4.9].

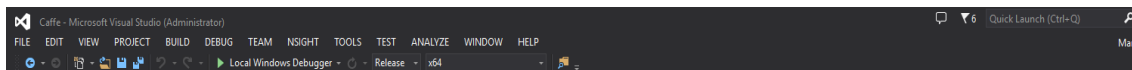


Obr. 4.8 Inštalácia OpenCV Default Build



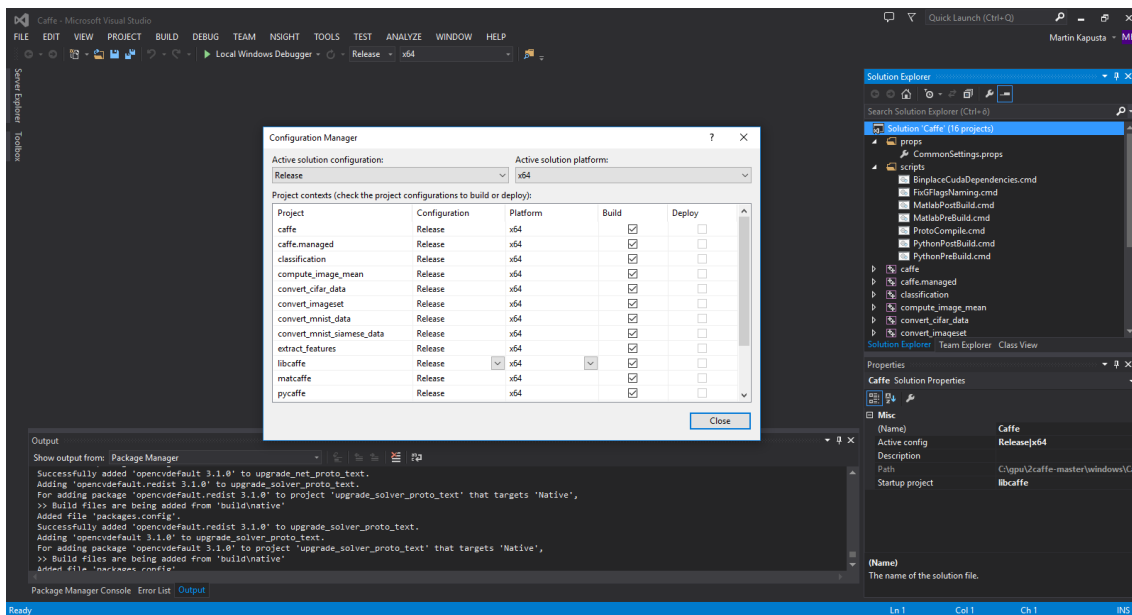
Obr. 4.9 Výsledok inštalácie

14. V hornej lište v *Solution configuration* sa musí prepnúť z „Debug“ na „Release“ a *Solution platform* nastaviť na „x64“. [Obr. 4.10]



Obr. 4.10 Ukážka nastavenia režimu

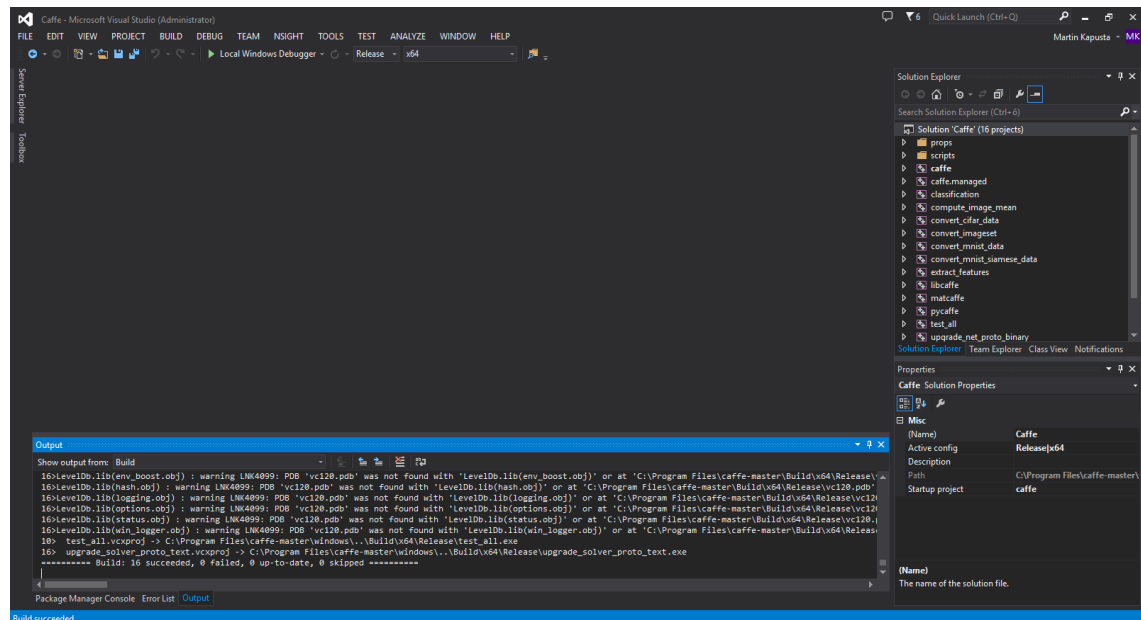
15. V tomto kroku sa len kontroluje nastavená konfigurácia kliknutím pravého tlačidla myši na *Solution 'Caffe' → Configuration Manager...*, otvorí sa tabuľka, v ktorej by všade malo byť *Configuration = Release*, *Platform = x64* a *Build* - všetko označené ako na obrázku [Obr. 4.11].



Obr. 4.11 Kontrola konfigurácie

16. Posledným krokom je kliknutie pravého tlačidla myši na *Solution 'Caffe' → Build Solution*. Keďže projekt je značne obsiahly, kompilácia trvá určitú dobu a výrazne zaťažuje procesor. Pri kompilácii sa objavia Upozornenia „Warnings“ typu „vc120.pdb was not found“ a podobné. Presne ich je 429 avšak tieto upozornenia nemajú vplyv na chod programu a vznikajú pri kompilácii statických knižníc. Statické knižnice sú k programu len pripojované, na rozdiel od

dynamických, ktoré sa s programom spájajú. Zostavovací program len nemôže nájsť konkrétne umiestnenie súboru. Výstup v konzole musí vyzerat' ako na [Obr. 4.12], kde je vidieť úspešnú kompiláciu všetkých šestnástich projektov.



Obr. 4.12 Výsledok kompilácie

4.4.3 Postup inštalácie bez použitia platformy CUDA

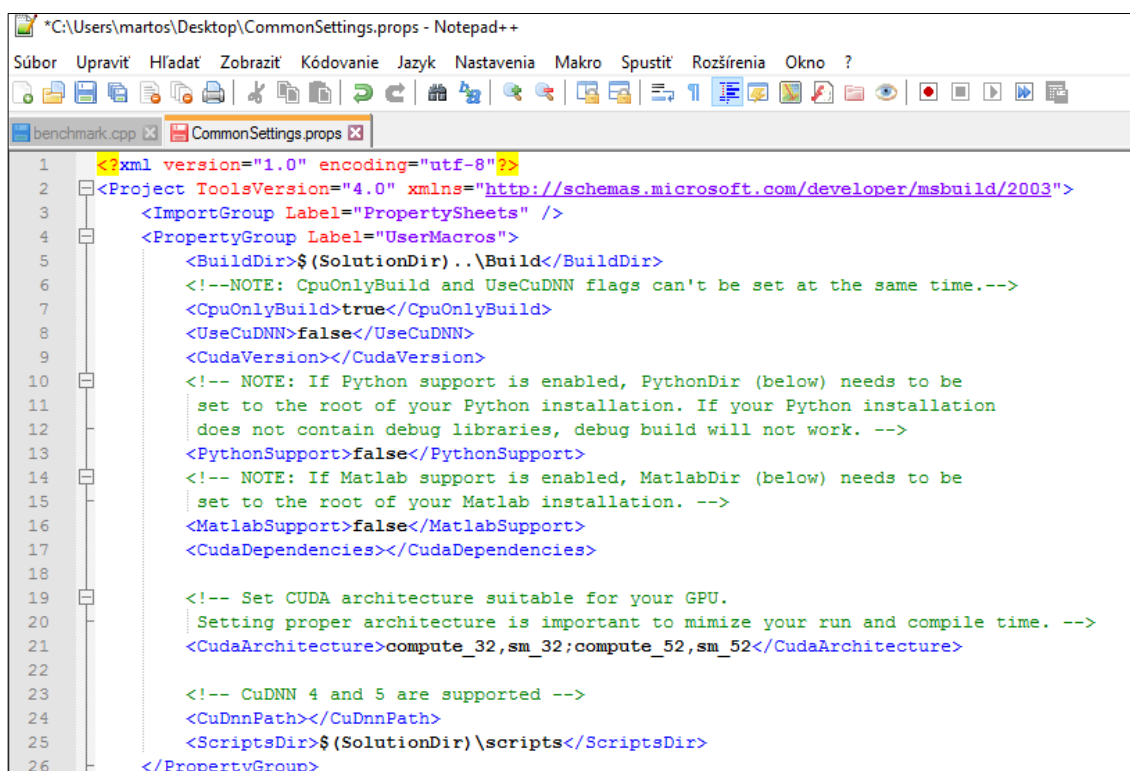
Postup inštalácie pre procesor je podobný ako inštalácia s použitím platformy CUDA. Pri inštalácii nie je potrebné túto platformu inštalovať a tiež sťahovať knižnice cuDNN. Verzia, ktorá beží iba na procesore, sa však v dnešnej dobe moc nepoužíva kvôli rýchlosti, pretože je potrebné spracovať veľké množstvo dát. Grafické karty dokážu paralelne spracovať väčšie množstvo informácií ako procesor. V návode sú uvedené iba kroky, ktoré sú odlišné oproti verzii s použitím platformy CUDA.

Prvý a druhý krok sú rovnaké ako v kapitole 4.4.2. Tretí a štvrtý krok nie je potrebné vykonávať, pretože bez grafickej karty od Nvidia sa tieto knižnice nedajú využívať. Piaty až siedmy krok sú rovnaké ako v kapitole 4.4.2. Podstatná zmena je v kroku číslo osem, kde sa budú meniť konfiguračné nastavenia tak, aby Caffe framework nevyhľadával knižnice cuDNN v počítači a nechcel sa spúšťať na

grafickom procesore. Budú sa upravovať niektoré riadky v súbore *CommonSettings.props*. Tento súbor bol vytvorený v predchádzajúcom kroku a nachádza sa v zložke „*caffe-master/windows*“. Otvorí sa pravým tlačidlom → Edit with Notepad++.

Nutné je upraviť riadok číslo sedem „<CpuOnlyBuild> false </CpuOnlyBuild>“. Hodnotu „false“ zmeniť na „true“. Tým sa zaručí, že Caffe framework bude pracovať na procesore (CPU).

Ďalej riadok číslo osem „<UseCuDNN>true</UseCuDNN>“ Hodnotu „true“ zmeniť na „false“. Tento krok je dôležitý, aby nevznikli chyby, že pri kompilácii by sa hľadali knižnice, ktoré sa v tomto prípade nedajú používať. Tiež je podmienkou, že hodnoty na riadku sedem a osem nemôžu byť rovnaké. Inak by program spadol. Skontrolovať treba, či na riadku číslo trinásť „<PythonSupport> false </PythonSupport>“ a riadku šesťnásť „<MatlabSupport> false </MatlabSupport>“ je nastavená hodnota „false“. Viac riadkov upravovať netreba. Súbor sa môže uložiť. Na obrázku je uvedený finálne upravený súbor *CommonSettings.props*. [Obr. 4.13]



```
1  <?xml version="1.0" encoding="utf-8"?>
2  <Project ToolsVersion="4.0" xmlns="http://schemas.microsoft.com/developer/msbuild/2003">
3    <ImportGroup Label="PropertySheets" />
4    <PropertyGroup Label="UserMacros">
5      <BuildDir>$(SolutionDir)..\Build</BuildDir>
6      <!--NOTE: CpuOnlyBuild and UseCuDNN flags can't be set at the same time.-->
7      <CpuOnlyBuild>true</CpuOnlyBuild>
8      <UseCuDNN>false</UseCuDNN>
9      <CudaVersion></CudaVersion>
10     <!-- NOTE: If Python support is enabled, PythonDir (below) needs to be
11       set to the root of your Python installation. If your Python installation
12       does not contain debug libraries, debug build will not work. -->
13     <PythonSupport>false</PythonSupport>
14     <!-- NOTE: If Matlab support is enabled, MatlabDir (below) needs to be
15       set to the root of your Matlab installation. -->
16     <MatlabSupport>false</MatlabSupport>
17     <CudaDependencies></CudaDependencies>
18
19     <!-- Set CUDA architecture suitable for your GPU.
20       Setting proper architecture is important to minimize your run and compile time. -->
21     <CudaArchitecture>compute_32,sm_32;compute_52,sm_52</CudaArchitecture>
22
23     <!-- CuDNN 4 and 5 are supported -->
24     <CuDnnPath></CuDnnPath>
25     <ScriptsDir>$(SolutionDir)\scripts</ScriptsDir>
26   </PropertyGroup>
```

Obr. 4.13 Finálne upravený súbor *CommonSettings.props*

Kroky číslo deväť až šestnásť sú rovnaké ako v kapitole 4.4.2. Výstup z konzole na obrázkoch sa bude líšiť, ale to na funkčnosť inštalácie nemá vplyv, pretože vždy sa čaká pokiaľ daný krok neprebehne až do konca.

4.5 Riešenie problémov pri inštalácii

Pri inštalácii môže vzniknúť problém s NuGet Packages, ktorý sa dá vyriešiť manuálnou inštaláciou tohto balíčka priamo zo stránky <https://www.nuget.org/>, kde sa nachádza aj návod a dokumentácia. Pre Caffé framework pod operačným systémom Windows s použitím Microsoft Visual Studio 2013 je potrebná verzia v2 nie najnovšia v3. Môžu sa inštalovať buď stiahnutím .exe súboru, alebo vpísaním príkazu do konzole priamo vo Visual Studiu, ale druhý spôsob nie vždy funguje. Túto možnosť sa odporúča použiť len v najnutnejšom prípade. Pokiaľ je to možné, použiť treba inštaláciu popísanú v kroku 11 z návodu v kapitole 4.4.2.

Niekedy je potrebné pridať *závislosti - dependencies* do projektu. Tento problém vzniká pri nesprávnom zadaní cesty k cuDNN v súbore *CommonSettings.props*. Pridávajú sa kliknutím pravého tlačidla myši na každý projekt zvlášť, zvoliť *Properties*→*VC++ Directories*→*Library Directories* a pridať cesty ku knižniciam, ktoré chýbajú. Väčšinou je to cesta k súboru cuDNN, stiahnutému v treťom kroku. Podobne sa musia podľa nutnosti pridať cesty k ostatným súborom v *Properties*→*VC++ Directories*→*Include Directories*. Po spustení testovacieho modelu nefungujú funkcie `TimerMilliseconds()`, `datum.Height()` a `datum.Width()`. Ostatné dôležité funkcie ako sú konvolučné vrstvy, algoritmy spätného šírenia chýb, ostatné výpočtové algoritmy, ktorých je veľké množstvo, fungujú bez problémov. Po diskusii s ľuďmi, ktorý Caffé framework implementovaný vo Visual Studiu alebo Matlabe či inom programe používajú som zistil, že tento test je orientačný. Pri tvorbe vlastného modelu všetko funguje bez problémov. Najdôležitejšie je, že sa projekt podarilo skompilovať bez chýb a zostavenie prebehlo úspešne pri všetkých šestnástich projektoch. Daný test je určený pre Linux. Caffé framework obsahuje spolu osemnásť projektov. Dva zvyšné projekty boli zostavovacím programom preskočené, pretože nepoužívame rozhranie Matlab alebo Python ale Visual Studio a ich kompilácia nie je potrebná.

5 NÁVRH NÁSTROJA PRE AUTOMATIZOVANÚ INŠTALÁCIU

5.1 Úvod k tvorbe inštalátora

Ďalším cieľom bakalárskej práce bolo vytvorenie inštalátora z postupu popísaného v kapitole 4.4 *Inštalácia po jednotlivých krokoch* a vytvorenie automatizovaného nástroja pre inštaláciu Caffé frameworku pre operačný systém Windows. Počet úkonov, ktoré sa očakávajú od užívateľa je obmedzený na minimum. Tým sa čo najviac znižuje riziko chýb, ktoré by mohli vzniknúť nesprávnym zadaním údajov alebo prehodením poradia krokov, čo by následne spôsobilo pád aplikácie. Aby nevznikli problémy pri zadávaní textového vstupu od užívateľa, sú nahradené funkciou pre výber zložky namiesto zadávania jej cesty do textového poľa.

Pre dokončenie inštalácie je vytvorený krátky video návod, ktorý zachytáva posledné štyri kroky. Tieto kroky sa nedajú vykonať pomocou externého nástroja, pretože sú previazané vo veľkej miere s Microsoft Visual Studio 2013. Medzi jednotlivými krokmi prebieha automatické sťahovanie rôznych balíčkov z internetu a inicializácia projektov. Bez otvorenia Visual Studia 2013 by mohli vzniknúť zbytočné chyby, ktoré by bolo ťažké opraviť. Preto je lepšie, aby sa vykonali priamo vo vývojovom prostredí. Inštalátor je vytvorený v dvoch verziách. Užívateľ si môže vybrať verziu s použitím platformy CUDA, ktorá beží na grafických kartách Nvidia alebo verziu bez použitia platformy CUDA. Verzia bez použitia platformy CUDA bude oveľa pomalšia, pretože všetky výpočty prebiehajú na procesore počítača (CPU) na rozdiel od verzie, ktorá beží na procesore grafickej karty, ktorý dokáže spracúvať paralelne väčšie množstvo informácií.

5.2 Realizácia inštalátora

5.2.1 Príprava súčastí programu

Pred tvorbou samotného inštalátora boli všetky potrebné knižnice, programy a súčasti umiestnené do jednej zložky a na záver vypálené na DVD disk. V jednotlivých súboroch sú vykonané potrebné zmeny. Medzi tieto zmeny patrí napríklad vytvorenie kópií súboru CommonSettings.props a jeho modifikácia pridaním textov `<!--text1-->`, `<!--text2-->`, `<!--text3-->`, `<!--text4-->`. Pomocné texty sú zakomentované aby neovplyvňovali funkčnosť Caffé frameworku. Na základe ich nájdenia v texte daný konfiguračný súbor upravuje inštalátor podľa požiadaviek užívateľa zadaných v aplikácii. Vymazané sú niektoré nepotrebné riadky kódu, ktorý sa automaticky nahrádza až po zvolení ciest inštalácie a verzie výpočetného výkonu grafickej karty. Účelovo sú pomenované niektoré zložky, aby inštalátor dokázal nájsť cesty ku konfiguračným súborom. Manuálne sú nastavené nepotrebné položky z „*true*“ na „*false*“ a tiež verzia platformy CUDA na 8.0, pri ktorej framework funguje. Platforma CUDA 8.0 je najnovšia vydaná verzia. Inštalátor platformy CUDA 8.0 je súčasťou aplikácie.

5.2.2 Programovací jazyk JavaFX

Pre samotnú tvorbu aplikácie bol zvolený programovací jazyk JavaFX. Vybraný bol, pretože poskytoval všetko, čo bolo pre realizáciu potrebné a tiež obsahuje pokročilé možnosti vytvárania graficky prívetivého užívateľského rozhrania, ktoré je pre inštalátor dôležité. Grafické rozhranie je vytvorené pomocou aplikácie JavaFX Scene Builder 2.0. Program je písaný vo vývojovom prostredí Eclipse Neon, ktoré bolo vybrané kvôli jeho vlastnostiam a predošlým skúsenostiam. Balíček s programom obsahuje triedu Main, ktorá obsahuje inicializačné metódy na spustenie grafického užívateľského rozhrania. Tiež je tu ošetrovaná možnosť, že si užívateľ po kliknutí na tlačidlo pre vypnutie danú akciu rozmyslí pomocou dialógového okna s výzvou, či chce inštaláciu prerušiť. Triedu Class, kde sa nachádzajú potrebné metódy a nastavené akcie pre jednotlivé tlačidlá, pomocou ktorých si užívateľ zadáva svoje

požiadavky pre inštaláciu a ďalej triedu GUI.fxml, v ktorej je uložené celé grafické užívateľské rozhranie.

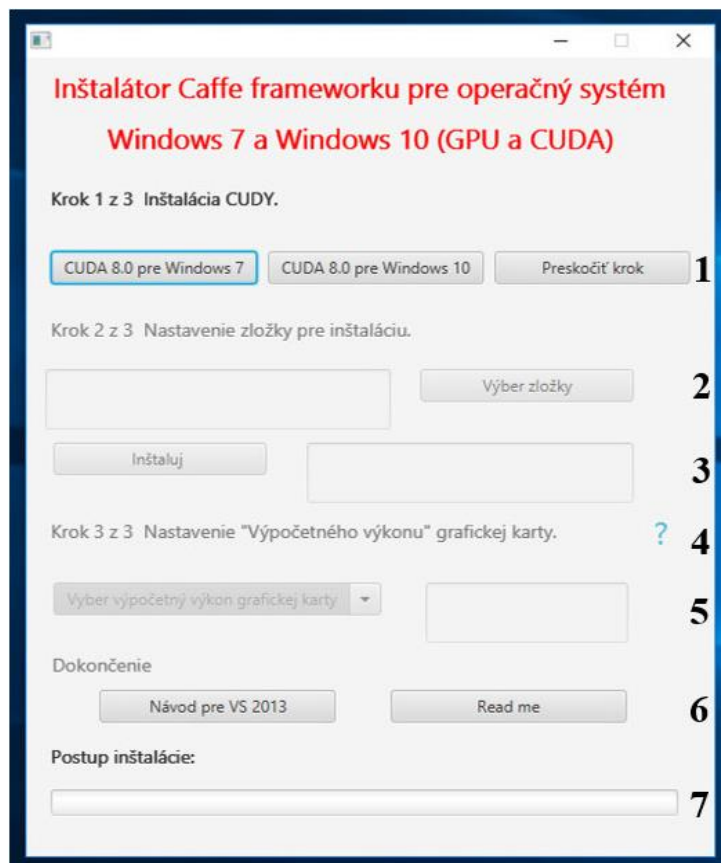
Inštalátor je vytvorený tak, aby sa nedali kroky, ktorých časová následnosť je dôležitá, preskočiť. Užívateľ v aplikácii zadá pokyn na nastavenie výpočetného výkonu grafickej karty, ale medzitým sa vykonajú aj iné kroky, o ktorých nevie, ale sú potrebné pre funkčnosť. Pri zámene poradia dvoch krokov by sa mohlo stať, že by metóda chcela upravovať konfiguračný súbor, čo by spôsobilo chybu `NullPointerException`, pretože by daný súbor zatiaľ neexistoval. Tieto chyby sú v programe ošetrené, ale napriek tomu je snaha v aplikácii zamedziť vzniku chýb blokovaním tlačidiel, ktoré nie sú potrebné pre vykonanie daného kroku. Vytvorené sú aj väzby medzi tlačidlami a prázdnyimi textovými poliami, aby si program nezobral ako absolútnu cestu prázdne pole, z ktorého by načítal cestu „null“, čo by spôsobilo pád programu a neúspešnú inštaláciu. V ďalších kapitolách sú popísané obidva vytvorené inštalátory Caffe frameworku.

5.3 Popis jednotlivých komponentov

5.3.1 Inštalátor Caffe s použitím platformy CUDA

Prvá verzia inštalátora je spojená s grafickými kartami od Nvidie, ktoré sú schopné pracovať s platformou CUDA a knižnicami cuDNN. Výstupom tohto inštalátora je nainštalovaný Caffe framework vo forme funkčného programu vo Visual Studiu 2013. Na počítači musí byť Visual Studio 2013 vopred nainštalované. V inej verzii daný framework nebude správne fungovať, pretože nové verzie nespôlupracujú s niektorými knižnicami pre hĺbkové učenie. Bližšie sú problémy s výberom verzií popisované v kapitolách pred návodom inštalácie. Sú to kapitoly 4.2 a 4.3.

Na nasledujúcom obrázku [Obr. 5.1] je zobrazené grafické užívateľské rozhranie aplikácie. Jednotlivé časti sú očíslované a pod obrázkom je postupne popisovaná ich činnosť. Inštalátor je zložený z krokov popísaných v návode, kapitola 4.4.2, na základe ktorých je vytvorená aplikácia tak, aby užívateľ nemusel zisťovať ďalšie informácie a bol schopný daný framework nainštalovať.



Obr. 5.1 Inštalátor Caffe s použitím platformy CUDA

Pri spustení aplikácie sa otvorí okno s názvom „Inštalátor Caffe frameworku pre operačný systém Windows 7 a Windows 10 (GPU a CUDA)“. V stave aktivované sú len tlačidlá potrebné pre prvý krok (1). V tomto kroku sa inštaluje platforma CUDA 8.0 pre verziu Windows 7 alebo Windows 10. Ak užívateľ platformu CUDA 8.0 už má nainštalovanú vo svojom počítači, klikne na tlačidlo „Preskočiť krok“. Týmto kliknutím sa deaktivujú tlačidlá z prvého kroku a aktivuje sa tlačidlo pre výber zložky.

Po kliknutí na „Výber zložky“ (2) sa otvorí dialógové okno, v ktorom sa vyberá zložka pre inštaláciu do počítača. Po výbere zložky pre inštaláciu sa aktivuje tlačidlo „Inštaluj“ (3). Kliknutím na tlačidlo „Inštaluj“ (3) sa začne kopírovanie potrebných súborov na zvolené miesto na disku. V textovom okne napravo od tlačidla „Inštaluj“ sa objavujú informácie o priebehu inštalácie a čas ako dlho trvalo kopírovanie. Táto funkcia je riešená pomocou vlákien, pretože grafické užívateľské rozhranie sa aktualizuje až po vykonaní všetkých krokov definovaných v akcii tlačidla. Po

úspešnom skopírovaní sa opäť deaktivujú tlačidlá, ktoré už užívateľ nebude používať.

Tretím krokom je nastavenie výpočetného výkonu grafickej karty. Napravo sa nachádza znak „?“ (4). Ak užívateľ nevie akú hodnotu nastaviť, klikne na tento znak a tým sa otvorí odkaz na stránku s tabuľkou grafických kariet so zodpovedajúcimi hodnotami výpočetného výkonu. Kliknutím na šípku v rozbaľovacom menu (5) sa objaví zoznam všetkých podporovaných hodnôt výpočetného výkonu grafickej karty, z ktorého si následne musí vybrať konkrétnu verziu.

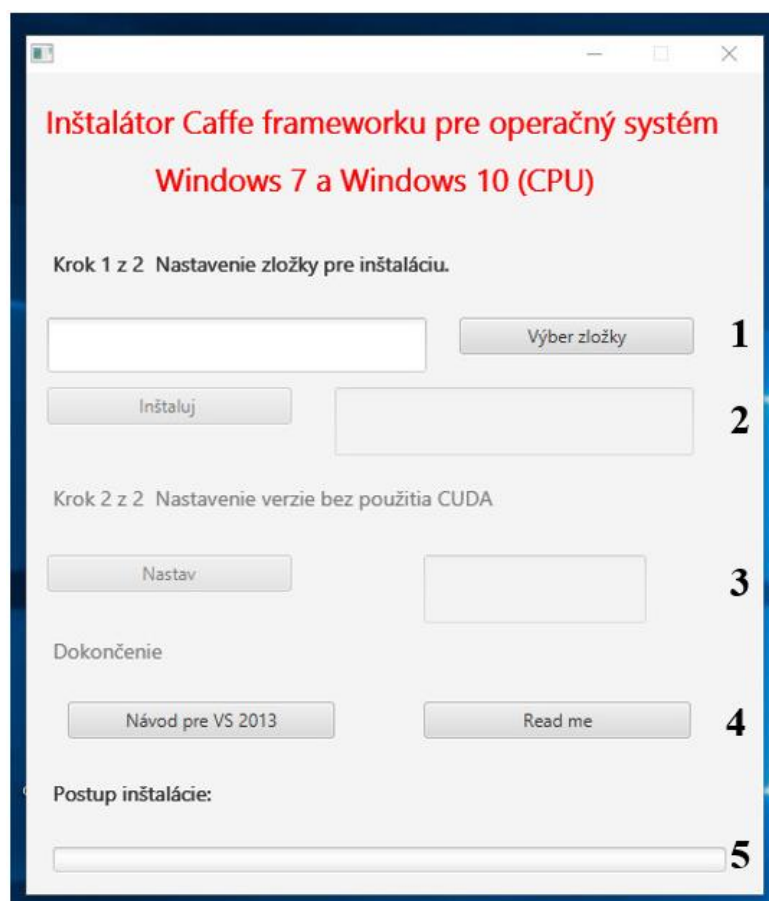
Ďalej sa tu nachádza tlačidlo „*Návod pre VS 2013*“ a „*Read me*“ (6). Po kliknutí na prvé spomínané tlačidlo sa otvorí video návod, kde je zobrazované ako realizovať posledné štyri kroky v Microsoft Visual Studio 2013 až po finálne zostavenie programu. Video návod je nahraný pomocou softvéru Bandicam. Následne je zostrihaný, pretože jednotlivé kroky zaberú určitý čas, prebieha sťahovanie balíčkov a inicializácia projektov. V textovom dokumente, po kliknutí na tlačidlo „*Read me*“ sú informácie o použití softvéru pri tvorbe inštalátora.

V sekcii „Postup inštalácie“ (7) je postupne zobrazovaný stav inštalácie.

Pri snahe o vypnutie inštalátora vyskočí dialógové okno. V tomto okne sa aplikácia pýta užívateľa: „*Chcete naozaj ukončiť inštalátor Caffé frameworku?*“. Podobným spôsobom je riešená situácia nastavovania výpočetného výkonu grafickej karty. Po výbere verzie aplikácia vyvolá dialógové okno. V tomto okne sa zobrazí dotaz s textom „*Naozaj chcete nastaviť výpočetný výkon grafickej karty na hodnotu xy ?*“. Tento spôsob bol zvolený kvôli tomu, že užívateľ zvykne najskôr kliknúť na tlačidlo, aj keď nevie presne čo nastaviť. Bez dialógového okna by sa hodnota musela nastaviť, pretože aplikácia by tlačidlá s týmto parametrom deaktivovala a nedalo by sa k nastaveniu vrátiť. Takýmto krokom by sa súbory Caffé frameworku museli manuálne zmazať a inštalácia znova spustiť od začiatku. Pri dotaze sa môže kliknúť na tlačidlo „*Nie*“ a vybrať inú hodnotu po prečítaní nápovede (4). Dialógové okná sú riešené pomocou metódy Alert().

5.3.2 Inštalátor Caffé bez použitia platformy CUDA

Táto verzia pracuje bez použitia platformy CUDA a tým pádom bez použitia grafickej karty. To znamená, že pracuje bez grafického procesoru. Všetky výpočty modelov, ktoré by sa používali budú realizované pomocou procesoru CPU (Central processing unit) a nie pomocou grafického procesoru GPU (Graphic processing unit) ako v prvej verzii inštalátora. Takýto spôsob učenia a trénovania konvolučných neurónových sietí zaberie viac času ako verzia pracujúca s použitím grafického procesoru. Podmienkou je nainštalované Microsoft Visual Studio 2013. Výstupom inštalátora je funkčný program implementovaný do Visual Studio 2013. Na obrázku je zobrazené užívateľské rozhranie Caffé frameworku bez použitia platformy CUDA. Jednotlivé časti sú očíslované a pod obrázkom [Obr. 5.2] sú postupne popisované.



Obr. 5.2 Inštalátor Caffé bez použitia platformy CUDA

Po spustení inštalátora sa objaví okno s názvom „*Inštalátor Caffé frameworku pre operačný systém Windows 7 a Windows 10 (CPU)*“. Na tento typ inštalácie nie je potrebná platforma CUDA. V rámci aplikácie ju neinštalujeme. Aktívne je zatiaľ iba tlačidlo „*Výber zložky*“ (1). Po kliknutí na toto tlačidlo sa objaví dialógové okno, v ktorom sa vyberá zložka, do ktorej chceme framework nainštalovať.

Po vybraní zložky sa aktivuje tlačidlo „*Inštaluj*“ (2). V textovom poli napravo sa nachádzajú informácie o stave inštalácie a celkový čas, ktorý trvala inštalácia. Po úspešnom skopírovaní súborov sa aktivuje pole pre nastavenie ostatných závislostí potrebných pre správnu funkčnosť.

Pri kliknutí na tlačidlo „*Nastav*“ (3) sa do konfiguračného súboru vpišu potrebné časti kódu, aby sa projekt skompiloval a pracoval bez grafického procesoru. Nachádza sa tu pole „*Dokončenie*“ v ktorom sú tlačidlá „*Návod pre VS 2013*“ a „*Read me*“ (4). Po kliknutí na prvé spomínané sa otvorí video návod so štyrmi krokmi na dokončenie inštalácie v Microsoft Visual Studiu 2013. Kroky sa nedajú realizovať pomocou externého nástroja, pretože ovplyvňujú množstvo súborov a mohli by vzniknúť neželané chyby a zbytočné problémy s kompiláciou.

Po kliknutí na tlačidlo „*Read me*“ sa otvorí textový dokument s popisom použitého softvéru pri tvorbe tejto aplikácie. V poli „*Postup inštalácie*“ (5) sa zobrazuje postup inštalácie po jednotlivých krokoch.

Pri snahe o vypnutie inštalátora vyskočí dialógové okno. V tomto okne sa aplikácia pýta užívateľa, či chce naozaj ukončiť inštalátor Caffé frameworku. Podobným spôsobom je riešená situácia nastavovania verzie bez použitia platformy CUDA. Po kliknutí na tlačidlo „*Nastav*“ (3) sa zobrazí dialógové okno s textom „*Naozaj chcete nastaviť verziu bez použitia CUDA?*“. Potvrdením tlačidla „*Áno*“ sú všetky súbory skopírované na zvolené miesto na disku. Nastavené sú tiež všetky závislosti a cesty ku konfiguračným súborom. V tomto momente je všetko pripravené na posledné štyri kroky v Microsoft Visual Studiu 2013.

5.4 Popis inštalácie a testovanie

Čas inštalácie pomocou inštalátora závisí od výkonu počítača a rýchlosti DVD mechaniky. Inštalácia platformy CUDA zaberie približne desať minút, kopírovanie súborov dve minúty. Počas kopírovania sa v textovom poli objaví informácia o tom, že prebieha inštalácia, aby si užívateľ nemyslel, že inštalátor prestal fungovať. Nastavovanie závislostí a verzie platformy CUDA alebo verzie bez jej použitia trvá približne dve sekundy.

Po dokončení inštalácie pomocou inštalátora je nutné prejsť na video návod a postupovať presne podľa pokynov. Video návod je zostrihaný. Sú z neho vystrihnuté kroky, keď sa projekty inicializovali alebo sa sťahovali súčasti z internetu. Z toho dôvodu je nutné medzi jednotlivými krokmi čakať, pokiaľ Microsoft Visual Studio 2013 nezrealizuje potrebné operácie. Potrebný čas závisí od výkonu počítača a rýchlosti internetového pripojenia. Pri testovaní na dvoch rôznych počítačoch, s operačným systémom Windows 7 a Windows 10, inštalácia zabrala približne štyridsať minút a prebehla bez problémov.

Inštalátor bol testovaný na dvoch počítačoch. Na prvom bol nainštalovaný 64 bitový operačný systém Windows 10, procesor Intel Core i7 3,1GHz, 8 GB RAM, grafická karta Nvidia Geforce 840M, výpočetný výkon 5,0. Na tomto počítači boli testované obe verzie aj s použitím platformy CUDA aj bez použitia platformy CUDA. Na druhom počítači bol nainštalovaný 64 bitový operačný systém Windows 7, procesor Intel Core i5 3,4 GHz, 8 GB RAM a grafická karta AMD Radeon HD 6950. Testovaná bola verzia bez použitia platformy CUDA. Vo všetkých prípadoch prebehlo zostavenie programu vo Visual Studiu 2013 úspešne bez hlásení „error“.

Aplikácia je napísaná v programovacom jazyku JavaFX. Je vo forme spustiteľného *.jar* súboru. Pre jeho spustenie je potrebné mať na počítači JRE 1.8 alebo vyššie. Aplikáciu je potrebné spúšťať ako správca, pretože pri prístupe na disk „C:/“ by jej mohol byť zamietnutý prístup. Tiež by sa nedal spustiť inštalátor platformy CUDA 8.0. Postup ako spustiť *.jar* súbor pomocou príkazového riadku je uvedený na disku s inštalátorom v textovom súbore „Čítaj pred inštaláciou“. Pri spúšťaní sa zadáva názov inštalátora zvolenej verzie s názvom DVD mechaniky

aktuálneho počítača. Presné príkazy s príkladom sú uvedené v textovom súbore „Čítaj pred inštaláciou“. Dôležité je zadať správne označenie aktuálnej DVD mechaniky, v ktorej je vložené DVD, inak nebude inštalátor pracovať z dôvodu nenájdenia ciest k súborom Caffé frameworku určených na kopírovanie a nebude schopný spustiť inštalátor platformy CUDA, video návod a ostatné potrebné súčasti.

Tiež som skúšal vytvoriť z vyexportovaného súboru *.jar*, súbor s príponou *.exe* pomocou programu *Launch4j*, pridaním *.manifest* súboru, aby sa spúšťala aplikácia ako správca. Nakoniec sa tento postup nedal použiť kvôli nefunkčnosti dialógových okien. Program bežal, pred spustením si vyžadoval správcovské práva, ale problém vznikol pri ukončovaní inštalátora alebo nastavovania verzie platformy CUDA. Preto bolo zvolené spúšťanie pomocou príkazového riadku v správcovskom režime, pri ktorom aplikácia funguje bez problémov.

6 ZÁVER

Bakalárska práca sa venovala problematike konvolučných neurónových sietí, ktoré sú špeciálnym druhom neurónových sietí. Téma konvolučných neurónových sietí je aktuálna a v posledných rokoch sa jej venuje čoraz väčšia pozornosť.

Na začiatku práce je popisované všeobecné použitie neurónových sietí v rôznych oboroch, história vzniku, biologický model neurónu a analogický matematický model umelého neurónu. Tak isto je popísaná biologická nervová sústava, z ktorej je odvodená umelá neurónová sústava používaná pre implemetáciu v rôznych programovacích jazykoch. Ďalej sú vysvetlené algoritmy pre dopredné a spätnoväzobné šírenie signálu, ktoré sú dôležité pre učenie a trénovanie neurónových sietí, popísaná je architektúra, rozdiely v učení s učiteľom a bez učiteľa, matematické vyjadrenie učenia konvolučných neurónových sietí a nakoniec ich praktické využitie v oblasti medicíny, automobilového priemyslu či zefektívnenia vyhľadávania obrázkov na Google, ktoré je presnejšie práve z dôvodu použitia tejto technológie.

Zdôvodnený je výber frameworku a vhodného vývojového prostredia pre implementáciu do operačného systému Windows 7 a 10 s potrebnými súčastami, knižnicami a platformami, čo bolo jedným z cieľov bakalárskej práce. Následne je vytvorený podrobný návod na inštaláciu Caffé frameworku pre operačný systém Windows s obrázkami a podrobným popisom jednotlivých krokov pre jednoduchšiu orientáciu v návode.

Navrhnutý je tiež nástroj pre automatizovanú inštaláciu Caffé frameworku pre Windows 7 a Windows 10. V rámci práce sú vytvorené dve verzie inštalátora. Popísané sú všetky prvky grafického užívateľského prostredia a ich funkcie. Zdôvodnený je výber programovacieho jazyka a vývojového prostredia pre tvorbu inštalátora a popísané triedy, z ktorých sa program skladá.

Výstupom je inštalátor Caffé frameworku v dvoch verziách. Prvá verzia je určená pre užívateľov s grafickými kartami od spoločnosti Nvidia, pretože používa platformu CUDA. Druhá verzia pracuje bez platformy CUDA, to znamená, že beží iba

na procesore CPU. Pomocou inštalátora je užívateľ schopný implementovať Caffé framework do vývojového prostredia Microsoft Visual Studio 2013. Užívateľ nemusí riešiť zdĺhavý postup diskusií na rôznych fórach a riešiť problémy spojené s inštaláciou pre Windows. Môže sa zamerať na konkrétne použitie frameworku a tvorbu vlastného modelu využívajúceho konvolučné neurónové siete pre tréning a učenie.

Pri inštalácii podľa návodu je dôležité dodržiavať popísané verzie jednotlivých súčastí, pretože frameworky, ktoré pracujú s konvolyčnými neurónovými sieťami sú primárne určené pre operačné systémy Linux, ktorých jadro pracuje iným spôsobom ako hybridné jadro operačného systému Windows, preto je ich implementácia do systému Windows náročná. Nutnosťou je nainštalované vývojové prostredie Microsoft Visual Studio 2013. Pri verzii s použitím platformy CUDA grafická karta, ktorá túto platformu podporuje. Obidve verzie si vyžadujú pripojenie na internet. Pri inštalácií pomocou aplikácie je potrebné mať v počítači JRE 1.8 (Java Runtime Enviroment) a 64 bitový operačný systém. Na DVD s inštalátorom sa nachádza Caffé framework s upravenými súbormi pre potrebu inštalácie. Na stránke, kde sa dá Caffé framework stiahnuť sa môžu časom objaviť drobné odlišnosti, keďže je to open source framework. Zmeny by nemali byť zvlášť rozsiahle, pretože Caffé framework nie je novinka a najväčšie chyby už boli odstránené.

V poslednej časti je popísaná časová náročnosť inštalácie s použitím vytvorených nástrojov pre inštaláciu testovanú na viacerých počítačoch a ich konfigurácia.

Práca by sa do budúcnosti dala rozšíriť práve kvôli tomu, že konvolyčné neurónové siete a oblasti umelej inteligencie spojené s učením počítačov sa dostávajú do popredia. Hľadajú sa spôsoby, ako umelú inteligenciu čo najviac priblížiť k schopnostiam človeka. Tento cieľ bol v určitých oblastiach už dosiahnutý avšak neustále je možné dosiahnuť lepšie výsledky a nájsť nové možnosti použitia konvolyčných neurónových sietí.

ZDROJE

[1] KRIESEL, David. *A Brief Introduction to Neural Network [online]. Bonn, Nemecko, 2005 [cit. 2017-05-16]. Dostupné z:*

http://www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-dkrieselcom.pdf. University of Bonn in Germany.

[2] ROBERTS, Eric. *Applications of neural networks. Neural Networks [online]. [cit. 2017-05-16]. Dostupné z:*

<https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Applications/index.html>

[3] RUZIC, Boris. *CLASSIFICATION OF ANIMAL SPECIES USING NEURAL NETWORK. Neuroph [online]. [cit. 2017-05-16]. Dostupné z:*

<http://neuroph.sourceforge.net/tutorials/zoo-part2/ClassificationOfAnimals-Part2.html>

[4] KAČENKA, Petr. *Neuronové sítě [online]. [cit. 2016-10-03]. Dostupné z:*

<https://mks.mff.cuni.cz/library/NeuronoveSitePK/NeuronoveSitePK.pdf>

[5] ROBERTS, Eric. *History: The 1980's to the present. Neural Networks [online]. [cit. 2016-10-03]. Dostupné z:*

<https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history2.html>

[6] KHAN ACADEMY *Overview of neuron structure and function. [online]. [cit. 2016-10-03]. Dostupné z:* <https://www.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/overview-of-neuron-structure-and-function>

- [7] TAUB, Ben. *Your Brain Connections Are As Unique As Your Fingerprint* [online]. [cit. 2016-10-03]. Dostupné z: <http://www.iflscience.com/brain/brain-connections-unique-fingerprint/>
- [8] TAUB, Ben. *New Map Of Brain Connections Is Most Detailed Ever Produced* [online]. [cit. 2017-05-16]. Dostupné z: <http://www.iflscience.com/brain/new-map-brain-connections-most-detailed-ever-produced/>
- [9] CELEBI, O.C. *Neural Networks and Pattern Recognition Using MATLAB: Chapter 7 Neural Network* [online]. [cit. 2016-10-03]. Dostupné z: https://www.byclb.com/TR/Tutorials/neural_networks/ch7_1.htm
- [10] HÁLA, Pavel. *Klasifikace spekter pomocí konvolučních neuronových sítí* [online]. Brno, 2014 [cit. 2016-10-03]. Dostupné z: https://is.muni.cz/th/323919/prif_m/paper.pdf Diplomová práce.
- [11] MEDERA, Dušan. *Klasifikácia chromozómov pomocou konvolučných neurónových sietí a inkrementálne prístupy k učeniu týchto neurónových sietí* [online]. Košice, 2007 [cit. 2016-10-03]. Dostupné z: <http://neuron.tuke.sk/medera/dizpraca/DizPraca.pdf>. Dizertačná práca.
- [12] STERGIOU, Christos a SIGANOS, Dimitrios. *NEURAL NETWORKS* [online]. [cit. 2017-05-16]. Dostupné z: https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#The%20Learning%20Process
- [13] GIBIANSKY, Andrew. *Convolutional Neural Networks* [online]. [cit. 2016-10-03]. Dostupné z: <http://andrew.gibiansky.com/blog/machine-learning/convolutional-neural-networks/>
- [14] BOUVRIE, Jake. *Notes on Convolutional Neural Networks* [online]. [cit. 2016-10-03]. Dostupné z: http://cogprints.org/5869/1/cnn_tutorial.pdf
- [15] WEYAND, Tobias, Ilya KOSTRIKOV a James PHILBIN. *PlaNet - Photo Geolocation with Convolutional Neural Networks* [online]. [cit. 2017-05-16]. Dostupné z: <https://arxiv.org/pdf/1602.05314.pdf>

- [16] ZHANG, Richard, Phillip ISOLA a Alexei A. EFROS. *Colorful Image Colorization [online]*. [cit. 2017-05-16]. Dostupné z: <http://richzhang.github.io/colorization/>
- [17] DAVID, Silver a kolektív. *Mastering the game of Go with deep neural networks and tree search [online]*. [cit. 2017-05-16]. Dostupné z: <https://www.nature.com/nature/journal/v529/n7587/full/nature16961.html>
- [18] *Deep Learning Frameworks [online]*. [cit. 2017-05-16]. Dostupné z: <https://developer.nvidia.com/deep-learning-frameworks>
- [19] GOMEZ-OL, Ricardo Guerrero. *Deep Learning frameworks: a review before finishing 2016 [online]*. [cit. 2017-05-16]. Dostupné z: <https://medium.com/@ricardo.guerrero/deep-learning-frameworks-a-review-before-finishing-2016-5b3ab4010b06>
- [20] JIA, Yangqing a Evan SHELHAMER. *Caffe [online]*. [cit. 2017-05-16]. Dostupné z: <http://caffe.berkeleyvision.org/>
- [21] *What is open source? [online]*. [cit. 2016-10-03]. Dostupné z: <https://opensource.com/resources/what-open-source>
- [22] *Notepad++ 7.4 [online]*. [cit. 2017-05-16]. Dostupné z: <https://notepad-plus-plus.org/download/v7.4.html>
- [23] *CUDA GPUs [online]*. [cit. 2017-05-16]. Dostupné z: <https://developer.nvidia.com/cuda-gpus>
- [24] *Neurón [online]*. [cit. 2017-05-16]. Dostupné z: https://sk.wikipedia.org/wiki/Neur%C3%B3n#/media/File:Neuron_slk.svg
- [25] *DEEPART.IO [online]*. [cit. 2017-05-16]. Dostupné z: <https://deepart.io/>

ZOZNAM SKRATIEK

CNTK - Microsoft Cognitive Toolkit framework - framework od firmy Microsoft

CPU - Central Processing Unit – centrálny procesor počítača

CUDA - platforma pre paralelné výpočetné aplikácie a aplikačné programovacie rozhranie vytvorené firmou Nvidia

cuDNN - The NVIDIA CUDA® Deep Neural Network library - grafickou kartou podporované knižnice pre hĺbkové učenie

DVD - Digital Video Disk – nosič s inštalátorom

GHz – GigaHertz

GPU - Graphics Processing Unit – procesor grafickej karty

GUI - Graphical User Interface – grafické užívateľské rozhranie

JRE – Java Runtime Environment – nainštalované prostredie potrebné na spustenie aplikácie napísanej v Java

LMS - Least mean Square – algoritmus

PCI - Peripheral Component Interconnect – zbernica k pripojeniu periférnych zariadení

RAM – Random Access Memory – operačná pamäť

RGB - Red Green Blue - farebný model

v3, v4, v5 - verzie 3, 4, 5

ZIP - formát súboru

60M - 60 000 000

ZOZNAM PRÍLOH

A OBSAH PRILOŽENÉHO DVD.....	57
------------------------------	----

A OBSAH PRILOŽENÉHO DVD

- 1** – upravené súbory Caffe frameworku pre inštaláciu s použitím platformy CUDA
- 2** – upravené súbory Caffe frameworku pre inštaláciu bez použitia platformy CUDA
- 3** - inštalátor platformy CUDA 8.0 pre operačný systém Windows 7 a Windows 10, Read me, videonávod – súbory potrebné pre funkčnosť aplikácie
- caffe_cpu_eclipse** – zdrojový kód verzie bez použitia platformy CUDA – súbory vývojového prostredia eclipse
- caffe_gpu_eclipse** – zdrojový kód verzie s použitím platformy CUDA – súbory vývojového prostredia eclipse
- CPU** – inštalátor verzie bez použitia platformy CUDA
- GPU** – inštalátor verzie s použitím platformy CUDA
- Čítaj pred inštaláciou** – manuál k spusteniu inštalácie pomocou príkazového riadku
- Konvoluční neuronové sítě na platformě Windows** – elektronická verzia bakalárskej práce